HZ BOOKS
华章教育

Pearson

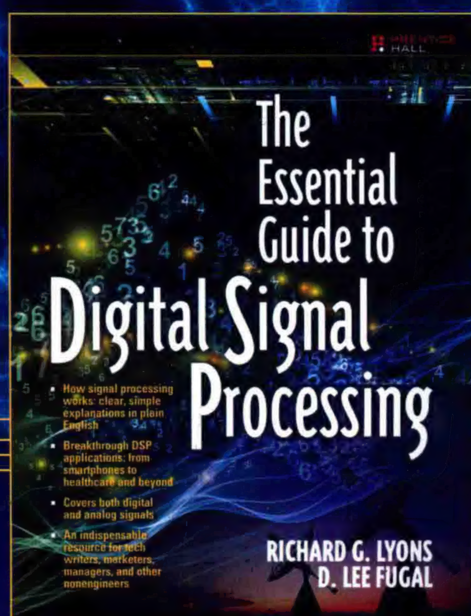
国外电子与电气工程技术丛书

数字信号处理精要

（英文版）

[美] 理查德 G. 莱昂斯（Richard G. Lyons）
D. 李·富盖尔（D. Lee Fugal） 著

The Essential Guide to Digital Signal Processing



机械工业出版社
China Machine Press

本书以最为平实简单的语言介绍数字信号、模拟信号和现代数字信号处理的应用；阐述了数字信号的采集、滤波、分析，以及数字信号处理在当下流行设备中的应用；是面向商业和非技术类从业人员的数字信号处理完全攻略。

本书主要内容

- 模拟信号的频谱、频率及其应用；
- 在现代电子设备中，数字信号的产生和使用；
- 数字信号处理应用的最新进展：从智能手机到保健产品；
- 什么是小波和小波变换如何应用在医学和智能手机等领域；
- 从自动音乐调谐软件到医学EKG信号分析等比较前沿的数字信号处理应用。

作者简介

理查德 G. 莱昂斯（Richard G. Lyons） Besser Associates公司咨询系统工程师和讲师，荣获IEEE信号处理协会2012年度教育家，曾在加州大学圣克鲁兹分校教授数字信号处理课程。作为前IEEE信号处理杂志副主编，他创办了“DSP Tips and Tricks”专栏。著有《Understanding Digital Signal Processing, Third Edition》（Prentice Hall, 2011）一书。

D. 李·富盖尔（D. Lee Fugal） S&ST技术咨询公司总裁和圣地亚哥IEEE信号处理协会主席，IEEE高级会员，同时也是IEEE的Third Millennium Medal获得者，拥有30多年工业界经验。他主要教授大学高年级的数字信号处理课程，并为全美的工程师提供AT短期课程。著有《Conceptual Wavelets in Digital Signal Processing》（Space & Signals Technical Publishing, 2009）一书。

Pearson
www.pearson.com

投稿热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn

封面设计·锡 彬

上架指导：数字信号处理
ISBN 978-7-111-56446-1
定价：49.00元

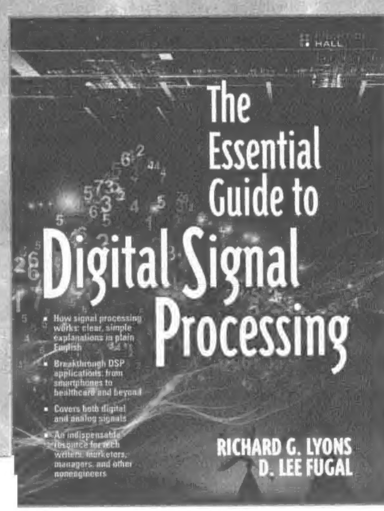
国外电子与电气工程技术丛书

数字信号处理精要

（英文版）

[美] 理查德 G. 莱昂斯（Richard G. Lyons）
D. 李 · 富盖尔（D. Lee Fugal） 著

The Essential Guide to Digital Signal Processing



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

数字信号处理精要（英文版）/（美）理查德 G. 莱昂斯（Richard G. Lyons），（美）D. 李・富盖尔（D. Lee Fugal）著. —北京：机械工业出版社，2017.4
（国外电子与电气工程技术丛书）
书名原文：The Essential Guide to Digital Signal Processing

ISBN 978-7-111-56446-1

I. 数… II. ① 理… ② D… III. 数字信号处理 IV. TN911.72

中国版本图书馆 CIP 数据核字（2017）第 065377 号

本书版权登记号：图字：01-2017-0901

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：谢晓芳	责任校对：殷 虹
印 刷：北京建宏印刷有限公司	版 次：2017 年 5 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16	印 张：12.25
书 号：ISBN 978-7-111-56446-1	定 价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991 88361066　　投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259　　读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，信息学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的信息产业发展迅猛，对专业人才的需求日益迫切。这对我国教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其信息科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀教材将对我国教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson、McGraw-Hill、Elsevier、John Wiley & Sons、CRC、Springer 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Alan V.Oppenheim Thomas L. Floyd、Charles K. Alexander、Behzad Razavi、John G. Proakis、Stephen Brown、Allan R. Hambley、Albert Malvino、Peter Wilson、H. Vincent Poor、Hassan K.Khail、Gene F. Franklin、Rex Miller 等大师名家的经典教材，以“国外电子与电气工程技术丛书”和“国外工业控制与智能制造丛书”为系列出版，供读者学习、研究及珍藏。这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也越来越多被实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着电气与电子信息学科建设的不断完善和教材改革的逐渐深化，教育界对国外电气与电子信息教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：（010）88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

HZ BOOKS
华章教育

华章科技图书出版中心

前　言

对于“信号”这个词，我们已经很熟悉了。所有承载信息的东西都可以称为信号，例如交通信号、求救信号，甚至还有烟雾信号。在纸牌游戏中，当我们拿到一手好牌的时候，往往尽量不给对方透露任何“信号”。那么，处理信号意味着什么呢？本书将采用现实生活中大家所熟悉的信号和信号处理方法，以最为简洁明了的方式解答这个问题。

大家可能没有意识到，实际上信号和信号处理时时刻刻影响着我们的日常生活。本书不仅展示了为什么信号和信号处理广泛存在于我们的日常生活中，而且更深入地解释了其中的原理。例如，为什么我们从收音机中听到的声音效果要远好于从手机中听到的声音效果？

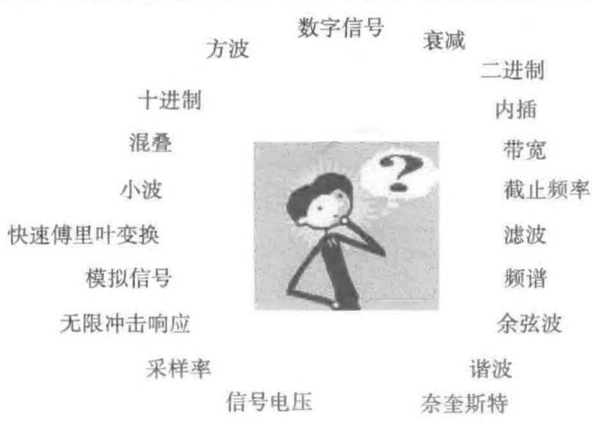
本书面向的读者是非技术类的人群，而不是工程专业的学生。因此，本书的主要目的有两个：首先，采用最少的数学公式，以通俗的方式阐述信号和信号处理的基本概念与原理；其次，介绍信号处理中的“语言”——术语。（为了便于阅读，本书在最后附上了完整的信号处理术语和缩写词。）

对于那些在公司制作或者使用信号处理硬件或软件的非技术类读者，本书将会是你的最佳选择。日常工作中，你可能会碰到许多看似神秘的概念和术语。本书将为你揭开这一层神秘的面纱，让你更深入地理解信号处理，从而更有效地与工程师或者其他技术类人员交流。

通常可以将信号分为两大类：模拟信号和数字信号。本书将逐步解释这两种信号的本质，以及如何在日常生活中使用它们来提高生活品质。

本书的章节是按照作者的理解安排的，你不必按照章节依次阅读，也不见得要阅读整本书。第 1 章主要介绍信号处理如何在近现代变得如此重要及其原因。第 2 ~ 5 章则阐述模拟信号和数字信号的基本性质。其他章节介绍模拟信号和数字信号的处理方法。

全书的内容大致如此。我们希望你能喜欢这本书，还望它能给你带来帮助。



Contents

目　　录

1 What Is Digital Signal Processing?

THE PHANTOM TECHNOLOGY

The technology of digital signal processing (DSP) has affected our modern lives in the most significant ways. If you watch television, connect to the Internet, use a digital camera, make a cell phone call, drive a car, type on the keyboard of a home computer, or use a charge or debit card, you are taking advantage of DSP. In fact, DSP is the technical *brains* in all those devices. Although we take advantage of DSP dozens of times a day, very few people have ever heard of digital signal processing and this strange situation is why DSP has been called a phantom technology. To show how much we depend on this *invisible* DSP technology, Table 1.1 provides a short list of what life would be like without DSP.

Table 1.1 Life without Digital Signal Processing

Missing Technology:	Comments:
No cell or smartphones.	No texting or Web surfing. Anyone remember pay telephones?
No computers.	No Internet, no e-mail, no Facebook, no YouTube, no Skype.
No cable or satellite television.	Viewing restricted to a few local, low-definition TV channels.
No compact discs (CDs).	Go back to audio cassette tapes.
No digital video discs (DVDs).	Who remembers movies on low-definition VHS magnetic tapes?
No charge card purchases	Cash or check only.

(Continues)

Table 1.1 *(Continued)*

Missing Technology:	Comments:
No digital cameras.	Plan on taking exposed camera film to the drug store.
No ultrasound and no MRI or CAT scans.	Revert to exploratory surgery (opening you up) to investigate internal medical problems.
No Global Positioning System (GPS).	Go back to paper maps.
No Doppler radar.	No long-range weather predictions.
No advanced oil exploration.	Higher gasoline prices. (Yes, even higher.)
No video games.	Kids would have to go outside to play.
No airline flights during bad weather.	Bring your sleeping bag to the airport.
No musical greeting cards.	How boring.

Given that we now realize how important DSP is in our daily lives, it's reasonable to ask just what is this thing called **DSP technology**. To understand the meaning of the phrase *digital signal processing*, we must first explain what we mean by the word *signal*.

WHAT IS A SIGNAL?

Any complete definition of the word *signal* must be, by necessity, somewhat vague. For example, some people define a signal as any representation of information conveyed to a receiver. Rather than discussing the meanings of those defining words, let's clarify what the word *signal* means to us by considering examples of signals that we've experienced in our daily lives. For example, when we listen to music produced by a loudspeaker, we're experiencing a signal in the form of sound waves traveling through the air that stimulates our eardrums. When we drive our cars to a traffic intersection, a light signal radiated by a red or green traffic light tells us whether we should stop or proceed. And if we ignore the red light, we find another red light following us down the road with a siren to *signal* us to pull over!

When you want to make a cell phone call, the symbol on your phone's screen, shown in Figure 1-1, is a visual indicator that your phone is receiving a sufficiently strong radio signal from a local cell phone tower. The height of a thermometer's mercury column is a visual signal indicating temperature. When we receive a kiss on the cheek, that's a tactile signal of affection. All of these examples are instances of receiving a *signal* that conveys information.

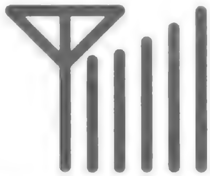


Figure 1-1 Cell phone signal-strength indicator.

ANALOG AND DIGITAL SIGNALS..............................

As it turns out, all signals fall into one of two major categories, *analog signals* and *digital signals*. The signals that we experience in our daily lives, the examples of sound and light signals mentioned in the previous section, are **analog signals**. Chapters 2 and 3 discuss analog signals in more detail.

Strangely enough, **digital signals** are nothing more than sequences of numbers. It's true—sequences of numbers that can be stored in the electronic memories of computers, digital cameras, and video game machines, or recorded on CDs and DVDs. Signal processing engineers have developed a way to convert analog signals, such as a sound or light signal, into digital signals (sequences of numbers). The digital signals contain *all* the information of the original analog signals. In addition, signal processing engineers have also developed the means to convert a digital signal back into an analog signal (sound or light). Converting an analog signal to a digital signal and then converting the digital signal back into an analog signal doesn't seem too useful, but that's where digital signal processing comes in.

DIGITAL SIGNAL PROCESSING ....

Digital signal processing is the mathematical manipulation of the numerical values of a digital signal that changes the digital signal in some advantageous way. For example, let's say a vocalist is singing into a microphone and we convert that analog voice signal to a digital signal. Next, the values of the numbers in the digital signal can be modified such that when the modified digital signal is converted back to an analog signal and played through a loudspeaker, we hear a slight echo in the singing that gives us a more pleasant sound. Manipulating pop singers' voices is standard operating procedure in today's music business. We discuss that topic in more detail in Chapter 4.

For a more serious example of digital signal processing, consider undergoing an electrocardiogram (EKG or ECG) test to check for problems with the electrical activity of your heart. Small electrodes, taped to your chest, detect an analog electrical signal produced by your heart that often looks like that shown in Figure 1-2(a). For various practical reasons, the analog electrode signal is contaminated with abrupt,

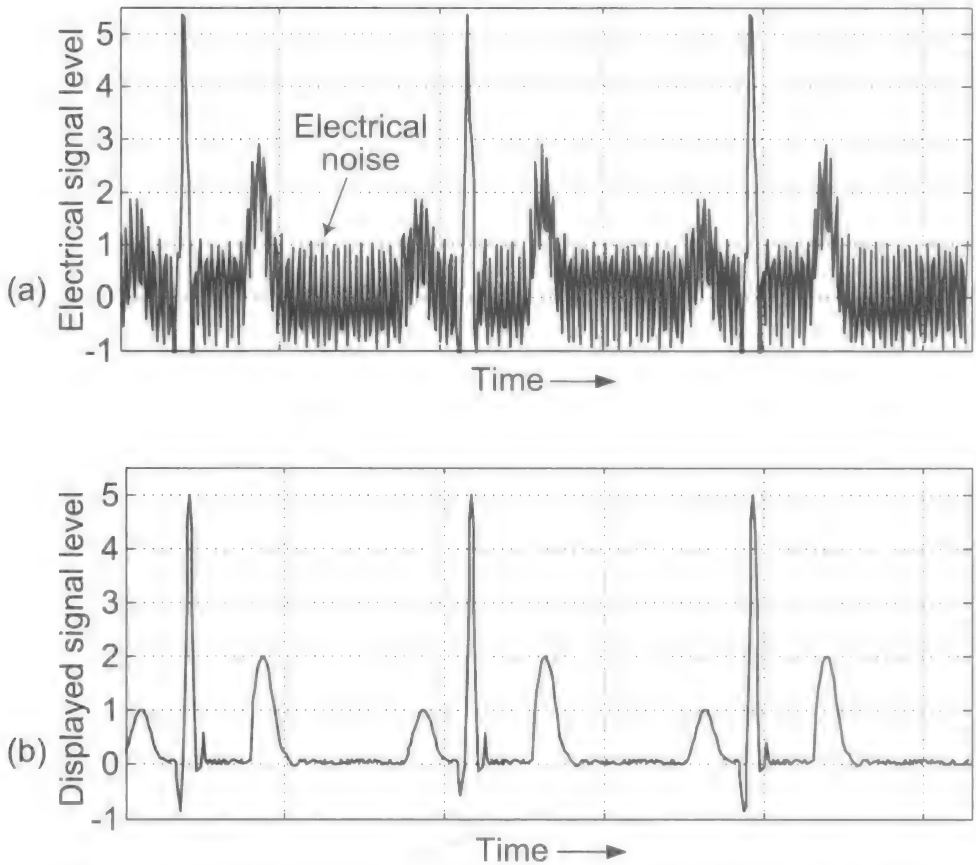


Figure 1-2 Electrocardiogram signals: (a) original measured noisy signal; (b) improved signal display after digital signal processing.

unwanted signal-level fluctuations, called **noise**, making it impossible for a doctor to evaluate your heart's electrical activity.

Today, digital signal processing comes to the rescue. As shown in Figure 1-3, the analog electrical sensor signal is converted to a digital signal. Next, the numerical values in the digital signal are modified in a way that eliminates the unwanted noise portion of the signal. The result is a clean EKG display, as shown in Figure 1-2(b), enabling a doctor to quickly evaluate the health of your heart

Other applications for DSP include military, industrial, space exploration, photography, communications, scientific, seismic, weather and many more. As we showed earlier in Table 1.1, life would go on without the benefits of DSP. However, we would have to do without so very many conveniences that we currently enjoy.

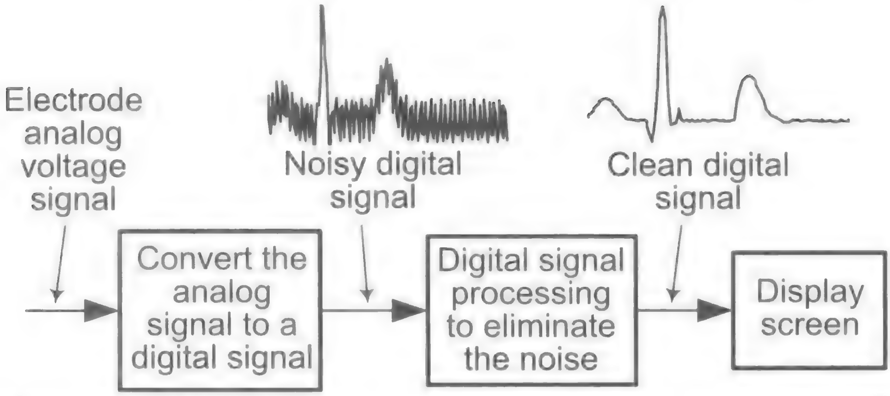


Figure 1-3 *Using digital signal processing to improve an electrocardiogram signal display.*

OK, this concludes our super-brief introduction to analog and digital signals, and digital signal processing. In later chapters, we'll learn more details about signals and signal processing.

WHAT YOU SHOULD REMEMBER ....

The concepts you should remember from this chapter are:

- We experience signals throughout our daily lives, usually in the form of analog sound and light signals.
- There is a way to convert analog signals, such as sound or light signals, into digital signals (sequences of numbers) that are stored in an electronic device. The digital signals contain *all* the information of the original analog signals.
- The numbers in a digital signal can be mathematically modified to improve some important characteristic of the signal, or reduce unwanted noise that contaminates it.
- The processed (modified and improved) digital signal can be converted back into an analog signal if necessary.
- The applications of DSP are many and varied. We may not always see where this phantom technology of DSP is used, but our lives would be very different without it.

2 Analog Signals

As we stated in the last chapter, the first step in understanding digital signal processing is to learn about analog signals. With that thought in mind, the goal of this chapter is to define and explain the nature of analog signals.

WHAT IS AN ANALOG SIGNAL?

For our purposes an **analog signal** is defined as any representation of a physical quantity that

- typically varies in value over time,
- has a value at all instants in time, and
- contains information.

Those characteristics seem a little mysterious but they're really not. You experience analog signals every day. The audio emanating from your cell phone speaker, the electrical video signal arriving from your television cable company, the height of the mercury column in an outdoor thermometer over a period of one day, and the fluctuating light intensity from a twinkling nighttime star are all examples of analog signals. An important aspect of analog signals is that they contain information that can be meaningful to us. With that said, let's look at a few analog signals in detail.

A TEMPERATURE ANALOG SIGNAL

As a simple example of an analog signal, the temperature curves in Figure 2-1 represent the high and low outdoor temperatures in Marquette, Michigan, over a period of one year. We can think of those curves as analog signals—they represent physical quantities

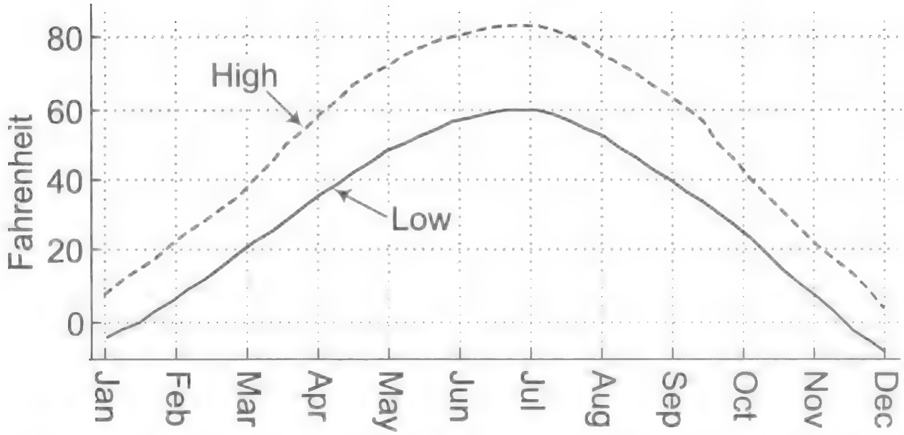


Figure 2-1 High and low outdoor temperatures in Marquette, Michigan.

that vary over time. For any given day of that year, we can look at the curves and estimate Marquette's high and low outdoor temperatures. And just what information do those Figure 2-1 analog signal curves contain? They tell us that if we're uncomfortable in cold weather we shouldn't accept a job offer in Marquette, Michigan.

A requirement for analog signals is that when drawn on a piece of paper, with time represented on the horizontal axis as shown in Figure 2-1, the tip of the pen or pencil never leaves the surface of the paper. There are no gaps, no missing information, in the curve. Such a curve can be called a *continuous curve*. In fact, many engineers refer to analog signals as **continuous signals**.

AN AUDIO ANALOG SIGNAL

Another example of an analog signal is an audio signal emanating from a **loudspeaker** such as that shown in Figure 2-2(a). The speaker's paper cone, in response to an audio electrical voltage applied to the speaker's electrical terminals, vibrates in and out. Figure 2-2(b) shows waves of air pressure fluctuations caused by the cone's vibration. The dark shaded bands on the right side of that figure indicate waves of high air pressure, with the intermediate white areas representing low air pressure.

When sound coming from a loudspeaker is a single audio tone, like the tone produced by a tuning fork, the fluctuating air pressure waves entering the listener's ears can be depicted as shown in Figure 2-2(c). When the loudspeaker cone moves outward, it compresses air molecules creating a small region of high air pressure. Then, when the loudspeaker cone moves inward, it creates a small region of low air pressure. Thus, the loudspeaker produces radiating regions of compressed and rarefied air molecules, waves of high and low air pressure that travel from the loudspeaker to the listener's ears.

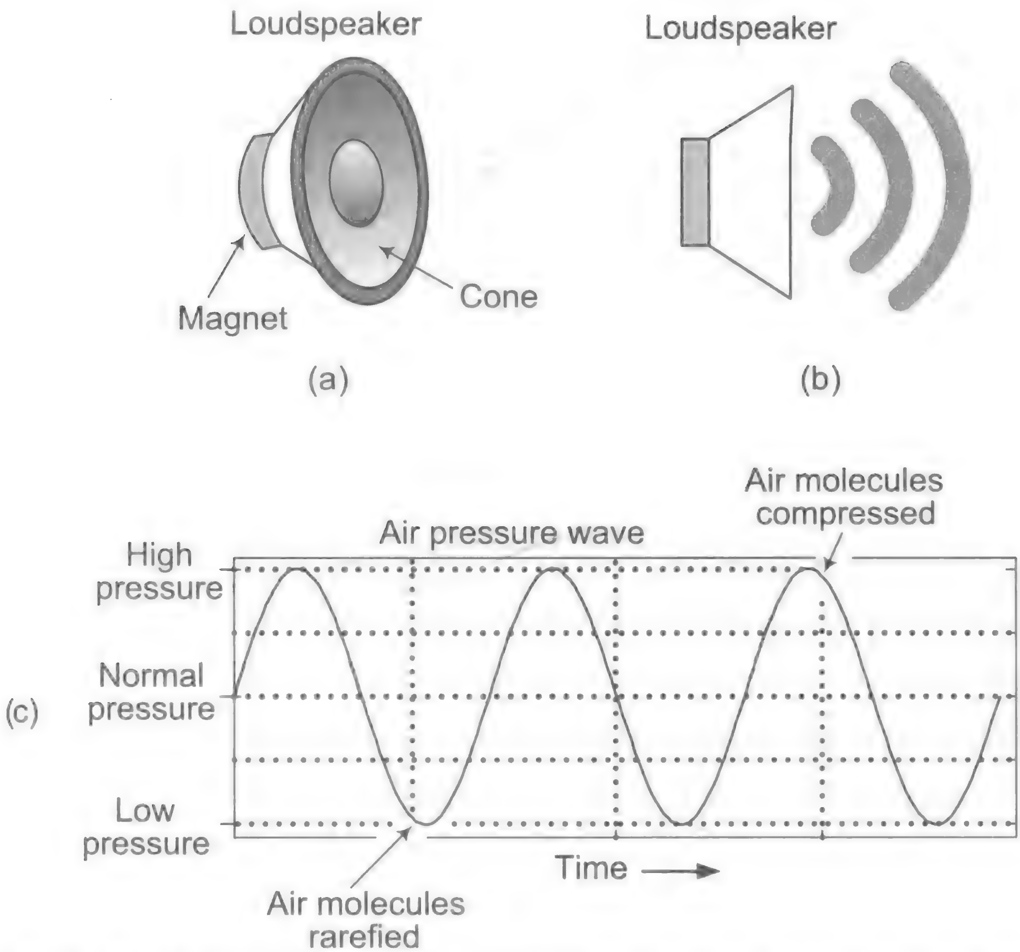


Figure 2-2 Audio tone analog sound wave: (a) loudspeaker; (b) loudspeaker and emanating air pressure waves; (c) tuning-fork wave of fluctuating air pressure entering a human ear as time passes.

Human ears are sensitive to these waves of fluctuating air pressure. When these waves enter our ears, an electrical signal is transmitted to our brains and only then do we experience the sensation of sound.

By the Way

To answer a popular question, if we define *sound* as traveling fluctuations in air pressure, then yes, a tree falling in the woods with no one nearby does indeed make a sound. If we define sound to be the electrical signal transmitted by the mechanisms of our inner ears to our brains, then no, that tree falling in the woods makes no sound.

AN ELECTRICAL ANALOG SIGNAL........................

As it turns out, engineers long ago invented a mechanical device that is sensitive to fluctuations in air pressure. When placed in the path of a traveling sound wave, the output of this device is a positive **voltage** when it is subjected to high air pressure and a negative voltage when it experiences low air pressure. This device is called a **microphone**. Engineers and technicians can view the analog voltage a microphone produces by connecting a microphone's output cable to an electronic instrument known as an **oscilloscope**, shown in Figure 2-3(a). The vertical axis on the oscilloscope's display represents voltage and the horizontal axis represents time.

If the tuning-fork sound wave in Figure 2-2(c) arrives at a microphone and the microphone's output cable is connected to the input port of an oscilloscope, then the scope's display would show the voltage signal as depicted in Figure 2-3(b). The voltage at any instant in time is called the **amplitude** of the voltage **waveform** and indicates its instantaneous energy. The peak positive voltage in Figure 2-3(b) is called the *peak amplitude* of the waveform. High peak-amplitude waveforms have more energy than low peak-amplitude waveforms.

The microphone's output voltage, representing fluctuations in air pressure, is an analog signal because it

- has an amplitude that varies as time passes,
- has a value at all instants in time (the voltage value changes smoothly in a continuous curve, with no gaps), and
- contains information such as peak-to-trough voltage difference and frequency (oscillations per second).

Again, the curve in Figure 2-3(b) is a voltage representation of the Figure 2-2(b) sound wave produced by a tuning fork. Thus, we can say that the Figure 2-3(b) voltage signal is analogous to, or the analogue of, the sound wave traveling in air. We referred to the curve in the figure as a fluctuating voltage. Because so many of the real-world analog signals that interest us take the form of analog voltage signals, let's pause for a moment and consider what is meant by voltage.

What Is an Electrical Voltage?

We can think of a voltage as a kind of electrical pressure having units of measure called *volts*. That electric pressure, given the opportunity, can cause electrical current (electrons) to flow. Figure 2-4(a) illustrates this notion with a simple flashlight battery, switch, and lightbulb circuit. There is a voltage of +1.5 volts at the positive end of the battery with respect to the negative end of the battery. (Likewise, there is a voltage of –1.5 volts at the negative end of the battery with respect to the positive end of

(a)

(b)

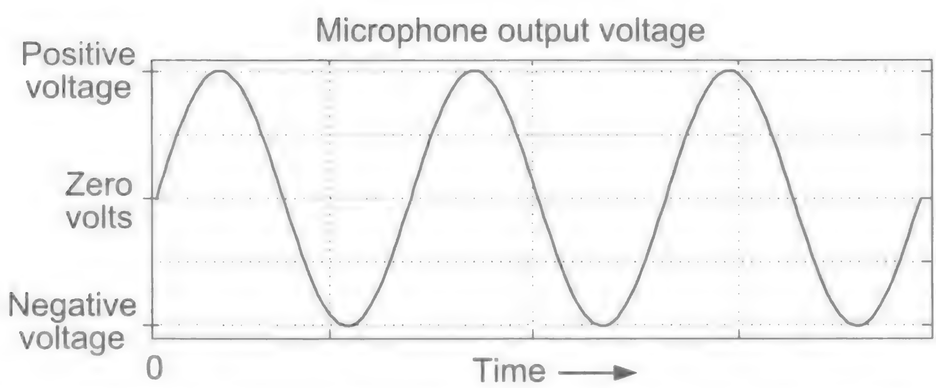


Figure 2-3 *Viewing a microphone's output: (a) a modern oscilloscope (courtesy of Tektronix Inc.); (b) displayed microphone output analog voltage waveform of a tuning-fork sound.*

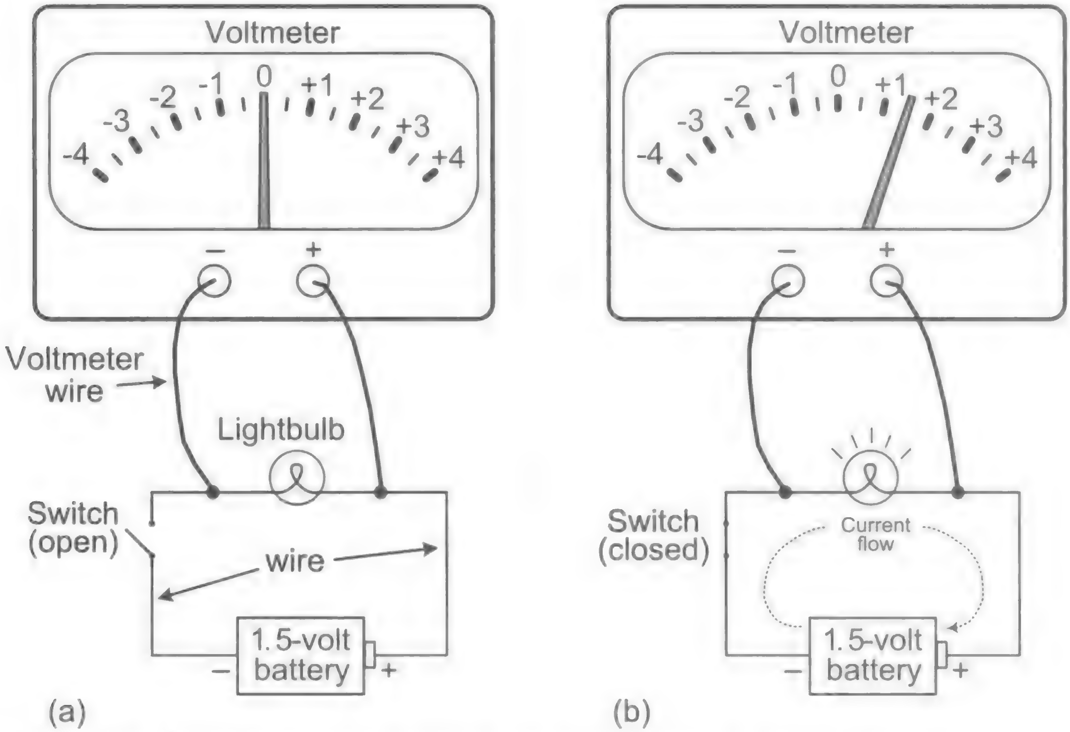


Figure 2-4 *Voltage across a lightbulb: (a) switch open; (b) switch closed.*

the battery.) In Figure 2-4(a), the switch is open (off), meaning that there is no path for electrons to flow through the wire from the negative terminal of the battery to the positive terminal of the battery. As a result, there is no voltage, zero volts, applied to the lightbulb and the bulb is not lit.

When the switch is closed, as shown in Figure 2-4(b), there is now an electrical path for electron current to flow through the wire clockwise from the negative terminal of the battery to its positive terminal, a +1.5 volt voltage is applied across the lightbulb terminals, and the bulb is lit. Specifically, electron current flows through the lightbulb and the bulb's filament becomes white hot, emitting visible light.

When the switch is closed, as in in Figure 2-4(b), the +1.5 volt voltage applied to the lightbulb is referred to as a **DC voltage**. The acronym DC stands for direct current, and for us DC voltage merely means a voltage that remains constant in value as time passes.

Let's consider another battery/lightbulb scenario. If we turned Figure 2-4's battery around, reversing its polarity, and closed the switch, then electron current would flow in a counterclockwise direction as shown in Figure 2-5. In that case, the voltmeter measures a negative –1.5 volts across the bulb's terminals. (Incandescent lightbulbs generate visible light regardless of the direction of current flow through their filaments.)

By the Way

The acronym DC is well over 100 years old. In the late 1880s, the pioneers in the field of generating electricity were in fierce competition over whether a constant **direct current** (DC, current with a flow in one direction only) or an **alternating current** (AC, current with a flow that alternates back and forth in direction) should be used to power the wonderful new invention called the lightbulb. Famous American inventor Thomas Edison vigorously campaigned on behalf of direct current. However, industrial giant George Westinghouse and his one-time employee Nikola Tesla were convinced that AC current, pioneered in Europe, was the best way to bring electricity to American homes. AC eventually won the battle because DC power generation stations could only supply power to buildings located within 1 mile (1.6 km) of the DC power station. The fluctuating nature of AC electrical power enables it to be transmitted over much longer distances by using the transformers located at the top of selected utility poles.

There are two concepts to learn from Figures 2-4 and 2-5. First, voltage is an electrical pressure that, when applied to some hardware component (the lightbulb), does work for us. If the lightbulb in Figure 2-4 were replaced by a low-voltage electric motor, then closing the switch would cause the motor to turn. This is how battery-operated screwdrivers work. Second, voltages have a polarity associated with them, as shown by the voltmeter readings in Figures 2-4(b) and 2-5.

Sinusoidal Wave Voltages

Let's consider a bit of terminology that we'll use later. The voltage waveform in Figure 2-3(b) is known as a **sinusoidal wave**, a generic term that can refer to either a sine or cosine wave, two waveforms we'll describe in the next sections. Sinusoidal waves are truly abundant in nature; here are some examples:

- waves in a river or ocean
- sound waves
- electromagnetic waves radiated by AM and FM broadcast stations
- seismic waves (earthquakes)
- starlight in the night sky

In fact, right now you're probably no farther than 10 feet from a voltage that fluctuates in a sinusoidal manner. We're referring to the oscillating voltage at the

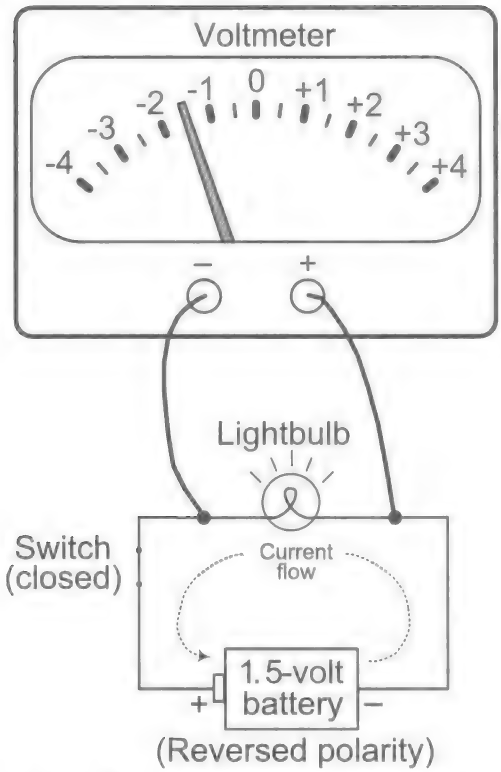


Figure 2-5 *Voltage applied to a lightbulb with reversed battery polarity and switch closed.*

conductors of your 120 volt AC (alternating current) wall socket. In Europe, it would be a 220 volt AC wall socket.

The voltage waveform in Figure 2-3(b) is a specific kind of sinusoidal wave called a **sine wave**, and sine waves show up in almost every aspect of digital signal processing. As such, we now take a closer look at sine waves.

Sine Waves

Let's engage in one of Albert Einstein's favorite pastimes, a thought experiment. Suppose an adventurous young lad decided it would be fun to climb out to the end of the huge minute hand of London's Big Ben and ride it around the clock's face for one hour. He might first want to climb out at 45 minutes past the hour when the minute hand is horizontal and pointing to the 9 position, as shown in Figure 2-6(a). Five minutes later at the 10 position, he would be lifted halfway (0.5) to the top (see Figure 2-6(b)). At

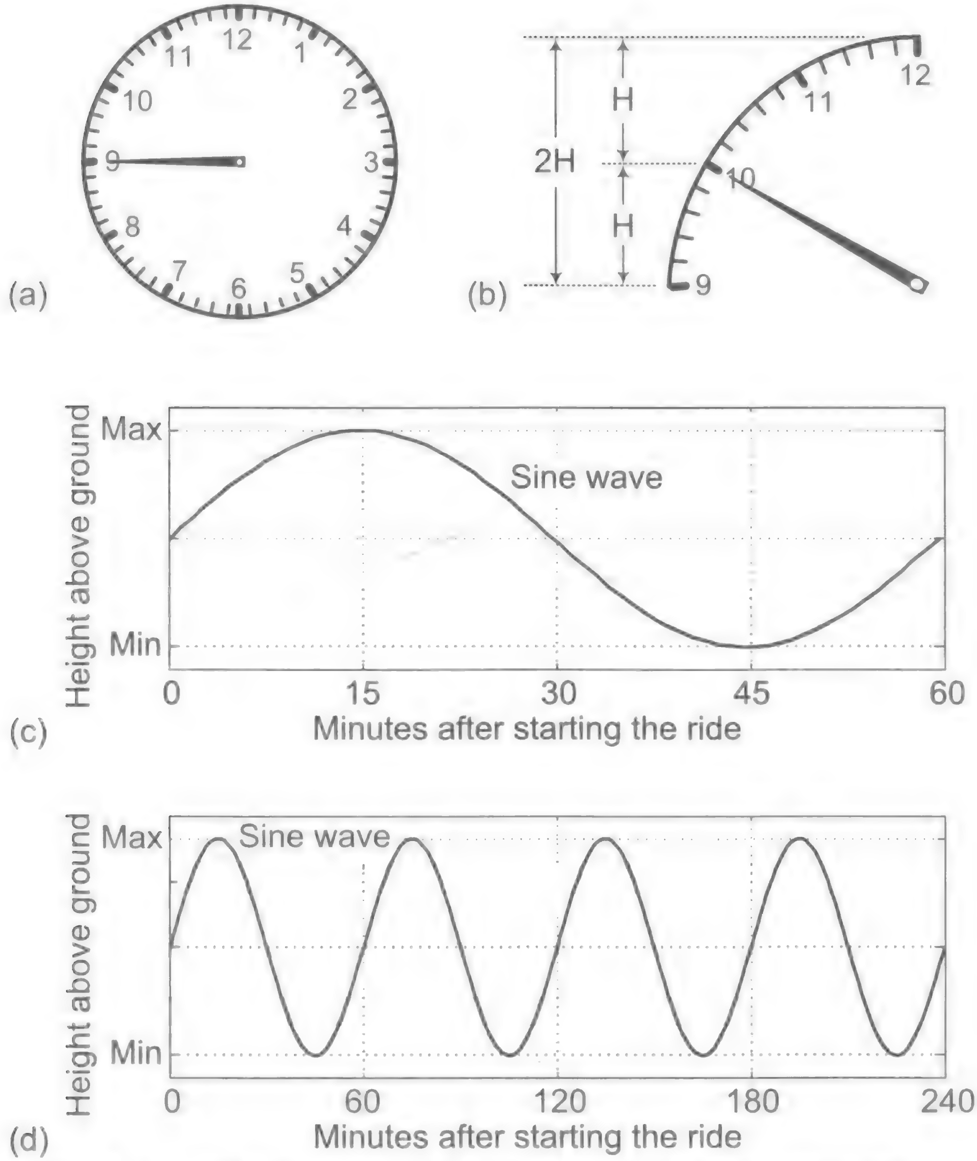


Figure 2-6 Riding Big Ben's minute hand: (a) starting point; (b) halfway point; (c) height versus one hour of time; (d) height versus four hours of time.

the 11 position, he would be most of the way toward the top (0.866), and at the 12 (top of the hour) position, he would be at the minute hand's maximum height.[1]

Continuing his ride, our daredevil would be at his original height when the minute hand is horizontal again pointing to 3, and at the minimum height at 6, the bottom of the hour. He would then finish his ride at the 9 position where he started.

If we were to sketch the relative height our young lad achieved during his hour of joy-riding, we would plot a single cycle of the well-known **sine wave** curve in Figure 2-6(c). This shows the important fundamental relationship between a sine wave and the vertical height of points around the perimeter of a circle. Figure 2-6(d) shows four cycles of a sine wave.

Cosine Waves

To continue our thought experiment: suppose our clock rider chose to wait 15 minutes (a quarter of an hour, a quarter *cycle*) and start his ride at the vertical 12 position rather than the horizontal 9. He would start and end his ride at the highest point above the ground and, if we sketched his relative height as time passed, it would look like the curve in Figure 2-7(a). That curve is called a **cosine wave**.

A sine wave is a delayed-in-time version of a cosine wave. Engineers call this relationship "a phase shift of one-quarter cycle between the sine and cosine," or in our case, one-quarter hour as depicted in Figure 2-7(b). Both sine waves and cosine waves are known as sinusoidal waves. Believe it or not, you've just learned the basics of sines and cosines!

Now here's how we adjust this knowledge to the world of engineering. The *ancients* used a circle rather than a clock to describe sines and cosines.[2] Instead of a *clockwise* rotating minute hand, they used a line 1 unit in length, representing the radius of a circle that rotates *counterclockwise* through 360 *degrees* around a circle, as shown in the two-dimensional Figure 2-8. For us, this unit-length line is represented by the bold arrow in the figure. The zero in the center of the circle represents a point where the x-axis value is zero and the y-axis value is zero.

The arrow starts by pointing to the right side at 0 degrees and proceeds counterclockwise around the circle 360 degrees back to where it started. During one arrow rotation, the vertical distance of the arrow tip above or below the x-axis is represented by the sine wave curve in Figure 2-6(c). The horizontal distance of the arrow tip to the right or the left of the y-axis is represented by the cosine wave curve in Figure 2-7(a). Those representations are shown in Figure 2-8(b).

---

1. That would be close to 200 feet above ground level for Big Ben on the Elizabeth Tower, nicknamed for its largest bell.
2. Among the ancients are Babylonians, Sumerians, and Greeks—not your authors.

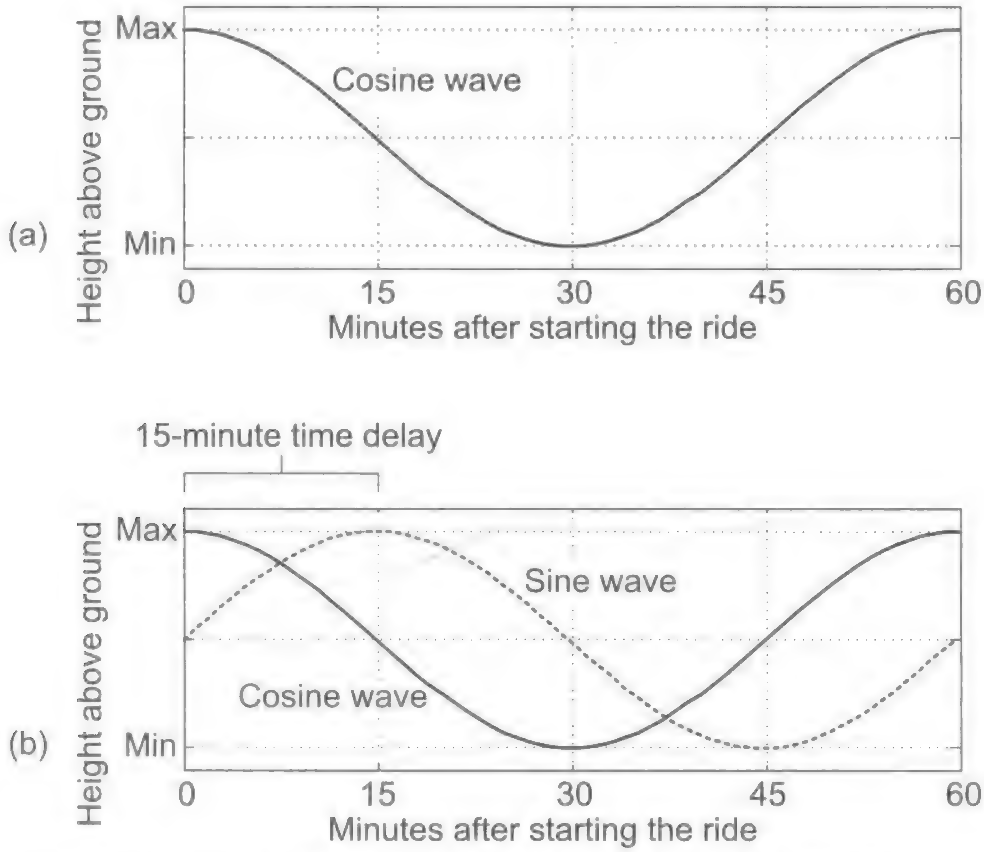


Figure 2-7 Riding Big Ben's minute hand starting at the 12 position: (a) height versus time; (b) time relationship between a sine wave and a cosine wave.

At this point, you might think that we've spent too much time here describing sine and cosine waves. To that we say sine and cosine waves pervade every aspect of both analog and digital signal processing, and time spent understanding sine and cosine waves is *never* wasted time. OK, with that said, let's explore the topic of voltages once more.

Sine and Cosine Voltages

Let's suppose we applied a sine wave voltage to the first input port of the four-channel oscilloscope shown in Figure 2-3(a) and applied a cosine wave voltage to the scope's

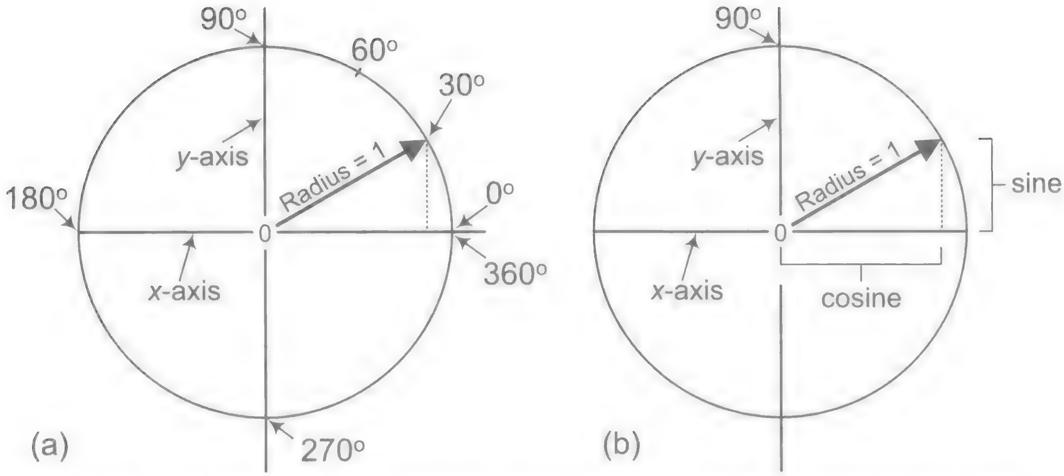


Figure 2-8 (a) Engineering basis for sines and cosines; (b) horizontal and vertical distance representations.

second input port. The scope's dual-trace display would appear as shown in Figure 2-9. There, we see that a cosine wave voltage is merely a delayed-in-time (by exactly one-fourth of a cycle) version of sine wave voltage. As it turns out, there are many signal processing applications that require the practitioner to generate both sine wave and cosine wave signals simultaneously. Sine and cosine waves are known as **periodic waves** because their waveforms periodically repeat themselves as time passes.

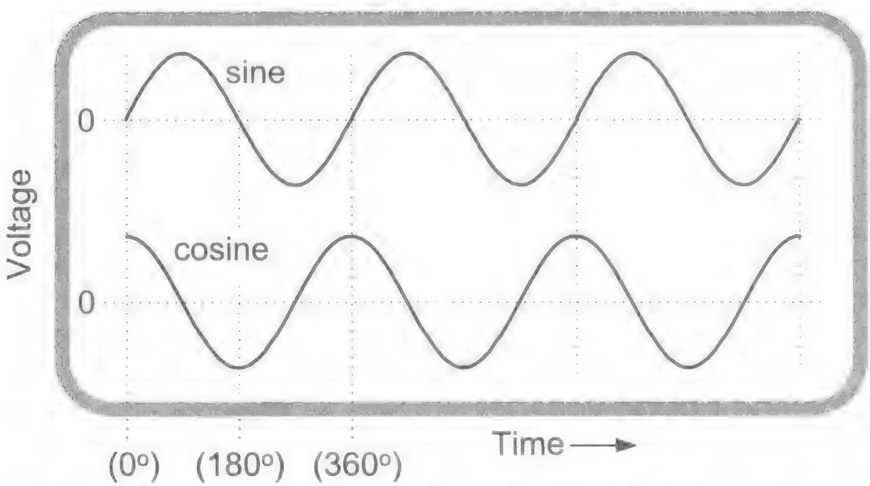


Figure 2-9 Oscilloscope display showing the time relationship between a sine wave voltage and a cosine wave voltage.

Other Useful Periodic Analog Signals

All sinusoidal wave voltages are periodic, but not all periodic voltages are sinusoidal. There are a number of specialized periodic analog voltage signals used by signal processing folk in various applications as well as for testing purposes.

Figure 2-10 shows what is called a **square wave** voltage that repeats its cycle once every half second; that is, two oscillations/second. We say that the **period** of that square wave is 0.5 seconds, meaning that it repeats its oscillation once every 0.5 seconds. Notice how that bi-level voltage very quickly fluctuates back and forth between two distinct voltage levels. As you can see, a square wave signal is not necessarily square in its shape; the voltage in Figure 2-10 is more rectangular than square. Nevertheless, when you hear someone talk about a square wave, what they actually mean is a signal that toggles between its two voltage levels and spends 50% of its time at each level. (The bottom two voltage waveforms on the oscilloscope in Figure 2-3(a) are square waves.)

For now, we'll call Figure 2-10 an analog square wave because it meets one of our most important definitions of an analog signal. Namely, when we draw the square wave signal on a piece of paper, the tip of our pen or pencil never leaves the surface of the paper. The square wave is *continuous*.

Inside your desktop and laptop computers, there is a square wave voltage called the **clock signal**. This constant-period square wave voltage is used to synchronize various circuit operations within your computer. When you read that a computer's clock frequency is 500 megahertz (Mhz), this means the computer generates and uses an internal square wave clock voltage oscillating 500 million times per second. The higher the clock frequency, the faster will be a computer's operating speed. In technical

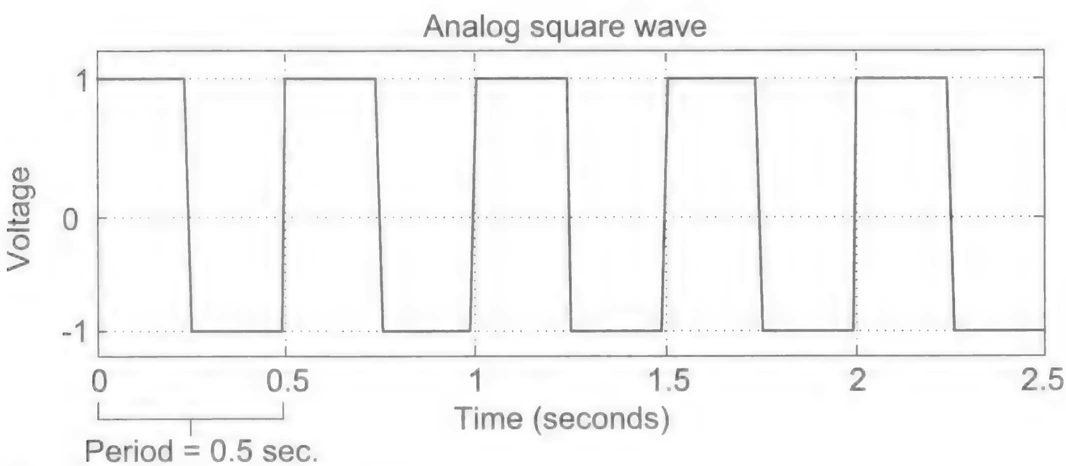


Figure 2-10 Analog square wave.

terms, a frequency of one oscillation per second is referred to as one **hertz** (Hz). We'll discuss the notion of *frequency* in more detail in the next chapter.

By the Way

In the late 1970s, the early, massed-produced Atari and Apple personal computers (PCs) had internal clock frequencies of roughly one megahertz (one MHz, one million oscillations/second). As of this writing, it's easy to buy a PC with a clock frequency of 4 gigahertz (4 GHz, four billion oscillations/second), an amazing increase in operating speed. If such a speed increase occurred in the automotive field since Henry Ford's early mass-produced Model T, current automobiles would have a top speed of 180,000 mph (288,000 km/h). That's more than 300 times faster than the cruising speed of a Boeing 747 jet airliner!

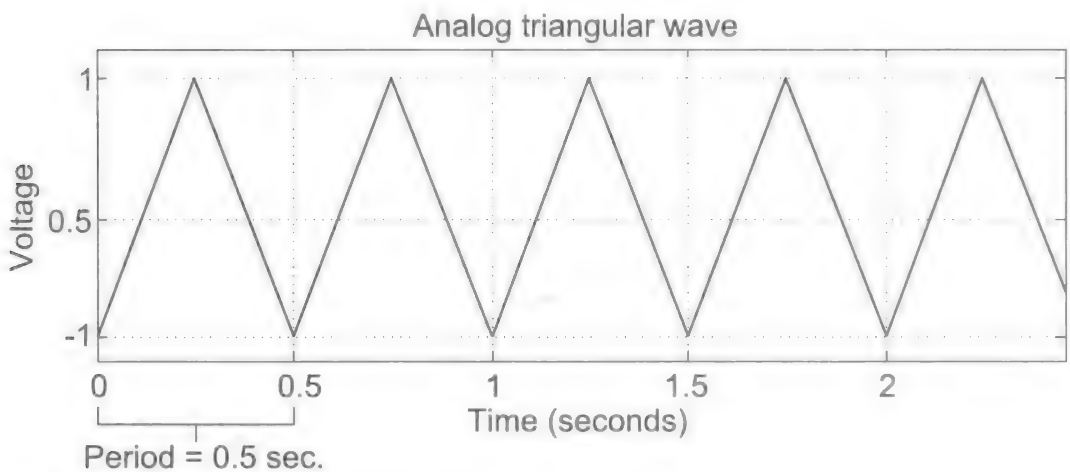


Figure 2-11 Periodic triangular-wave analog signal.

Now and then, you may hear signal processing engineers speak of a **triangular wave** voltage. This is a periodic voltage waveform as shown in Figure 2-11. It's easy to see why that signal is called a triangular voltage.

We'll encounter square and triangular waves again in the next chapter. Now that we're somewhat comfortable with the meaning of the word *voltage*, we'll return to the topic of analog signals.

A HUMAN SPEECH ANALOG SIGNAL

To expand our knowledge of analog signals, let's consider Figure 2-12(a), which shows the voltage output of a studio microphone when Capt. James T. Kirk, of the starship *Enterprise*, speaks the words "Mister Spock." That voltage waveform is rich in its fluctuating complexity. Obviously, the waveform in Figure 2-12(a) is not the sound of the actual words. Instead, the waveform is a voltage analogue of their sounds. If we amplified this voltage and connected it to the terminals of a loudspeaker, then we'd hear the actual audio sound of the words "Mister Spock." In this scenario, the

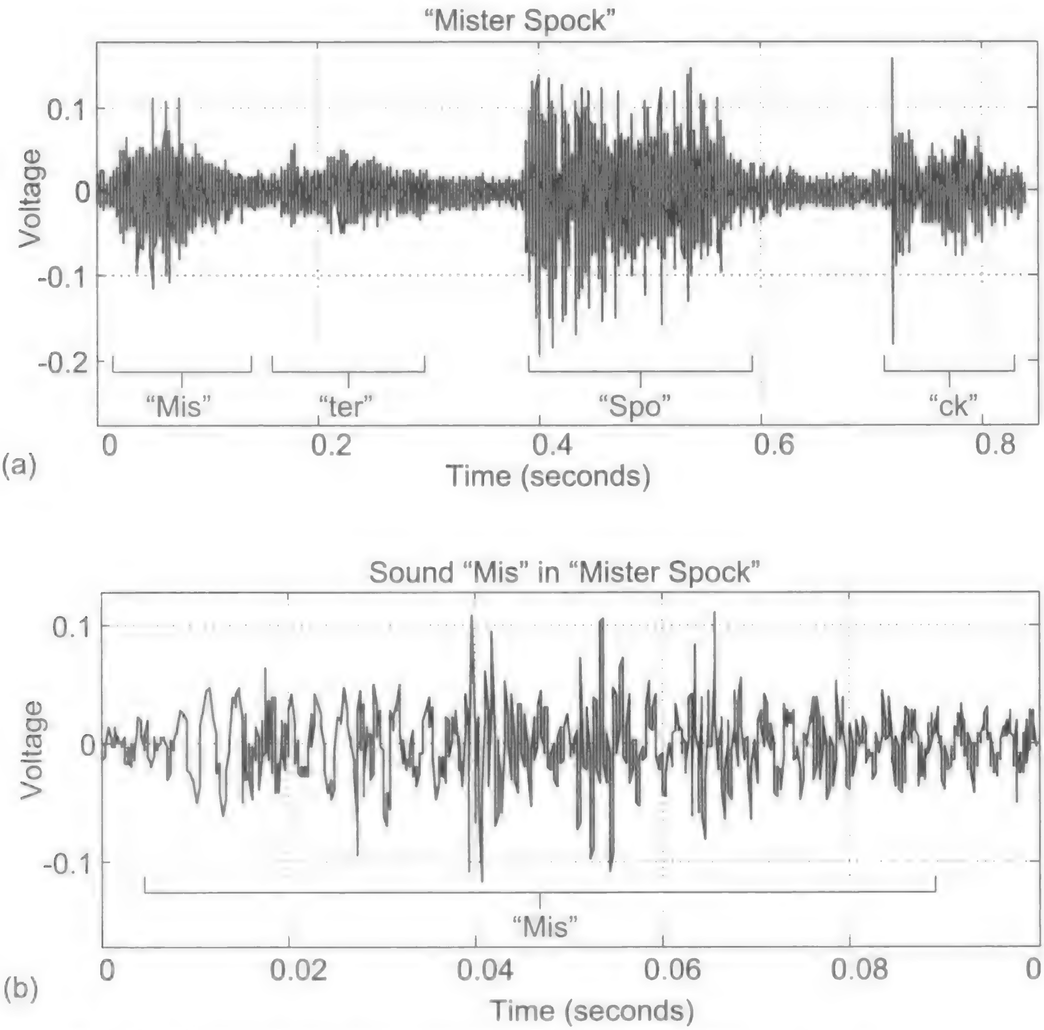


Figure 2-12 Analog audio speech signal: (a) "Mister Spock"; (b) "Mis."

fluctuating voltage causes the loudspeaker's cone to oscillate in and out, generating traveling waves of high and low air pressure. Our ears convert those air pressure fluctuations to an electrical signal sent to our brains, and we experience the sensation of hearing Capt. Kirk's voice.

We can examine the complicated pattern of the analog voltage in Figure 2-12(a) further by zooming in on the first syllable of the word "Mister." Figure 2-12(b) shows the details of the voltage waveform of just the syllable "Mis." Signal processing people work with voltage waveforms like this every day. They collect, examine, characterize, and sometimes modify these sorts of analog signals. In later chapters, we'll learn about the useful operations that can be performed on analog signals.

By the Way

Audio engineers sometimes combine multiple audio analog signals to create a new audio signal. For example, in the 1998 movie *Godzilla*, in the scene where the beast is crashing its way through New York City, the sound of collapsing buildings was added to the sound of attacking helicopters. The term used to describe the process of adding multiple audio signals is called mix down, or just plain **mixing**. (No, audio mixing engineers are not called mixologists. That title is reserved for another profession.)

WHAT YOU SHOULD REMEMBER ....

Wow! We covered a lot of technical material in this chapter. If you've plowed your way through this entire chapter, pat yourself on the back because you've learned an astounding amount of analog signal processing theory.

You may think we belabored the subject of analog signals a bit too much in this chapter, especially for a book about digital signal processing. Don't worry, we're not wasting time, paper, or ink. All of the concepts we learned about analog signals will help us understand digital signals.

The concepts you should remember from this chapter are:

- Analog signals are information-carrying physical quantities that vary in value as time passes.
- Physical quantities such as speed, vibration, temperature, sound pressure, and light intensity can be converted to voltages on a cable, allowing them to be viewed on the screen of an oscilloscope.

- Signal processing engineers work primarily with analog signals in the form of time-varying voltages.
- Periodically varying voltages, like sine and cosine wave voltages, are very common in the world of signal processing.
- The repetition rate, per unit time, of a periodic voltage is called frequency.
- Frequency is most commonly measured in units of hertz (Hz). One Hz is equal to one cycle per second.

3 Frequency and the Spectra of Analog Signals

In the last chapter, we looked at graphical plots of how analog signals vary in value (fluctuate) as time passes. Those plots are crucial in understanding the nature of any given analog signal. However, there's another very useful, and revealing, way to describe an analog signal. That alternate description is called the **spectrum** of an analog signal. Although you don't often encounter the concept of spectra in your everyday life, signal processing people study and analyze signal spectra every day. When engineers examine (analyze) an analog signal, knowing a signal's spectrum is as important as knowing the temperature, blood pressure, and heart rate of a patient being examined by a medical doctor.

So, in our study of signal processing, it's important that we learn what a signal's spectrum is and why such information is useful. That's the goal of this chapter. But before we discuss just what the spectrum of an analog signal is, we must, by necessity, briefly return to and focus on the notion of **frequency**.

FREQUENCY

In the last chapter, we introduced the term *frequency*, a term that we now need to define in a more formal way.

Frequency is a characteristic of periodic (repetitive) events. For us, frequency is the measure of how often some repetitive event occurs over some period of time. That is, frequency is a number of cycles (repetitions) per some unit of time. For example, when the needle of the tachometer on an automobile dashboard points to the number 2, that means the automobile's engine is rotating at a repetition rate of 2,000 revolutions per minute (RPM). Each revolution is a repetition, making the frequency 2,000 repetitions per minute.

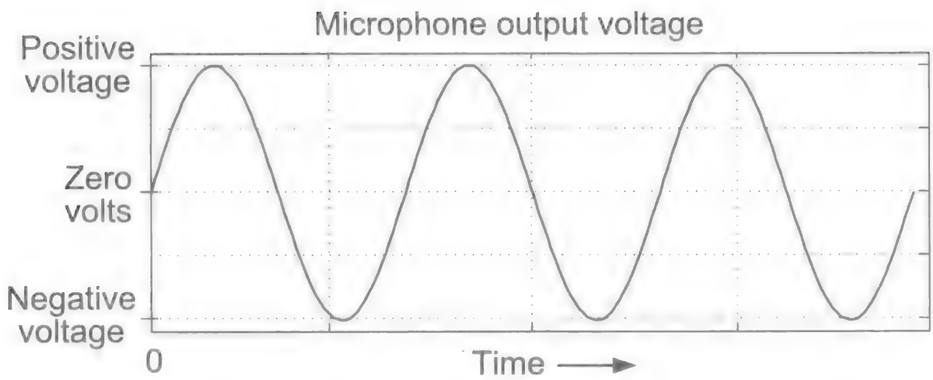


Figure 3-1 Microphone output analog voltage of a tuning-fork sound.

Cycles per Second

In the last chapter, we discussed the idea of pinging a tuning fork and the fork's oscillating sound wave arriving at a microphone. If the microphone's output cable were connected to the input port of an oscilloscope, then the scope's display would show the sinusoidal voltage signal depicted in Figure 3-1. For example, a tuning fork can be used to define the standard pitch of the A key on a piano keyboard (the A key above middle C). This standard pitch is called "A440." Thus, in Figure 3-1 the frequency of those sinusoidal analog wave oscillations is 440 hertz (440 oscillations per second, 440 cycles per second). For brevity, engineers write 440 hertz as 440 Hz.

Another frequency you're probably familiar with is the frequency of the AC (alternating current) electrical power at a wall socket in your home. Depending on your geographical location, the AC power line frequency will be either 50 Hz or 60 Hz. For example, in the United States, if you look at the bottom panel of any appliance on your kitchen counter, you'll see the characters "60 Hz."

A short list of power line frequencies is given in Table 3.1. AC power line voltages are sinusoidal in shape, meaning that the voltage of the *hot* conductor relative to the *neutral* conductor fluctuates from positive to negative, and then back to positive, just as does the voltage waveform in Figure 3-1.

Table 3.1 AC Power Line Frequencies

Location	Frequency
North and Central America	60 Hz
Europe, China, India, Africa	50 Hz
South America, Japan	both 50 Hz and 60 Hz

By the Way

In the early days of electricity, the unit of measure for frequency was **cycles per second** (cycles/second). That's why the words "kilocycles/second" and "megacycles/second" are printed on the tuning dials of old radios to indicate frequency. In the 1960s, the European and North American scientific communities adopted the word *hertz* (Hz) as the unit of measure for frequency in honor of the German physicist, Heinrich Hertz, who first demonstrated radio wave transmission and reception in 1887.

The single-syllable word *hertz* was a good choice because it's easy to say and it sounds plural in English. We're fortunate that Hertz's last name was nothing like Arnold Schwarzenegger's.

Signal processing engineers work with signals that cover an astounding range of frequencies. Audio engineers process analog signals in the frequency range of 20 Hz to 20 kilohertz (20 kHz, 20,000 Hz). Radio communications engineers deal with analog radio waves with frequencies in the range of kilohertz up to thousands of megahertz. Some cell phones operate at 900 megahertz. The frequency of the radiation inside the typical microwave oven is roughly 2.5 gigahertz (2,500 megahertz). Astronomers monitor analog stellar radiation with frequencies measured in terahertz (trillions of hertz). Table 3.2 gives the language and notation, and Figure 3-2 presents a crude graphical depiction, of various frequencies encountered in modern technology.

In case you haven't seen it before, the notation in the center column of Table 3.2 is called **scientific notation**, a convenient and precise way for engineers and scientists to write very large and very small numbers. A review of that method for writing large numbers is provided in Appendix A.

Incidentally, the lowercase "k" in kHz is not a typographical error. For many decades, kilohertz has been abbreviated as kHz and megahertz has been abbreviated MHz with the uppercase "M." That's just the way it is.

Table 3.2 Frequency Nomenclature

Frequency	Scientific Notation	Nomenclature
1,000 Hz	10^3 Hz	kilohertz (kHz)
1,000,000 Hz	10^6 Hz	megahertz (MHz)
1,000,000,000 Hz	10^9 Hz	gigahertz (GHz)
1,000,000,000,000 Hz	10^{12} Hz	terahertz (THz)

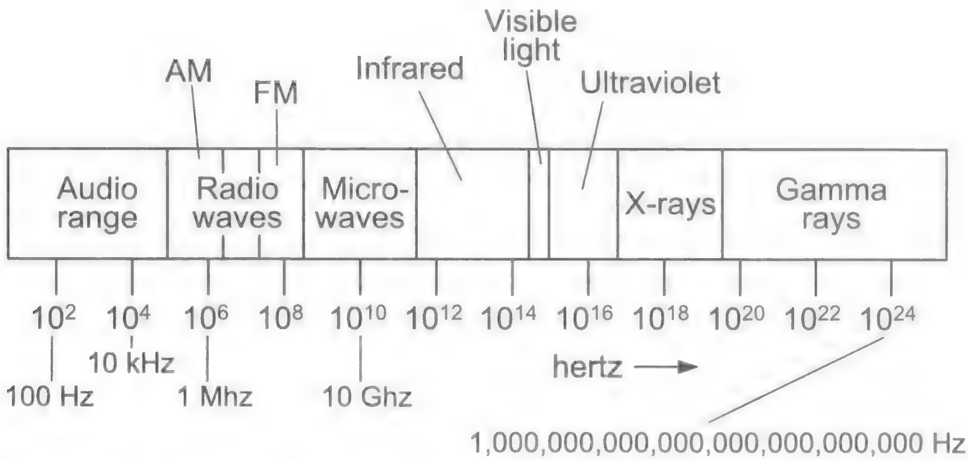


Figure 3-2 Range of frequencies encountered in modern technology.

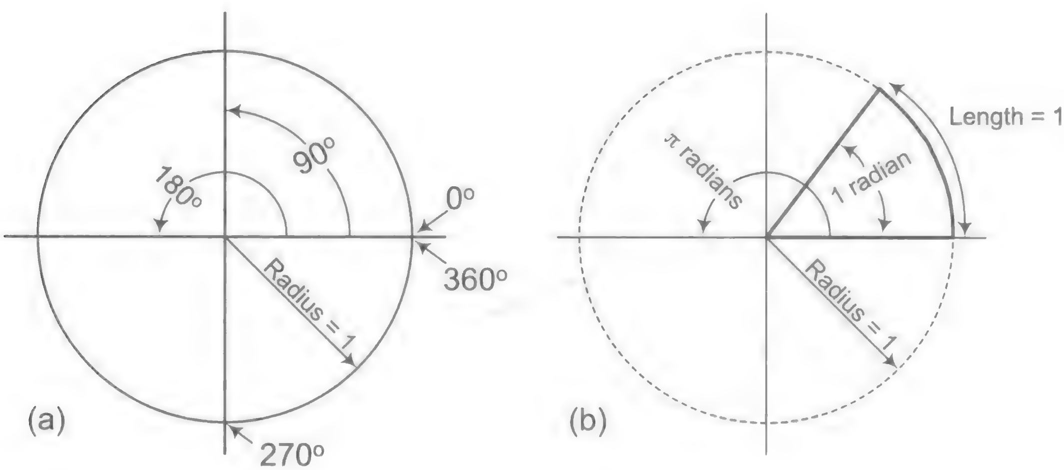


Figure 3-3 Angles in a circle: (a) 360°; (b) 2π radian.

Radians per Second

Occasionally you'll hear an engineer refer to a sine wave's or a cosine wave's frequency in terms of **radians per second**. It's straightforward to explain why people sometimes use such terminology.

Maybe you remember from geometry that the length of the circumference (perimeter) of a circle is π (pi) times its diameter.[1] Since the diameter is twice the radius,

1. $\pi = 3.14159\ldots$ is one of the fundamental constants in our universe. Its decimal digits extend to infinity to the right of its decimal point. But for us, we'll just say that $\pi = 3.14$.

the circumference of a circle can also be expressed as 2π times the radius length. To travel once around a circle takes 360 degrees, as shown in Figure 3-3(a). Likewise, to go once around the circle takes 2π radians, sometimes abbreviated as just 2π. So 360 degrees (one cycle) is equal to 2π radians. Therefore, a radian is an angle equal to $360°/2\pi$, roughly 57.3°, as shown in Figure 3-3(b).

Again, from geometry we think of a circle as containing 360 degrees, but in their mathematical equations it's much more convenient for engineers to think of a circle as containing 2π radians. In any case, if an engineer says a "sine wave's frequency is 6280 radians/second," that's Geek Speak for $6280/2\pi = 1{,}000$ Hz (cycles per second).

THE CONCEPT OF SPECTRUM

So far in this chapter and in Chapter 2, we've looked at graphical representations of a few different fluctuating analog signals. We viewed two-dimensional graphs where the vertical axis of a graph represents a signal's instantaneous amplitude (energy) and the horizontal axis represents time, as we saw in Figure 3-1. Such graphs, which show how a signal's voltage amplitude level changes as time passes, are called **time-domain** plots. The horizontal axis of time-domain plots is always in units of time. Another powerful, and important, way to characterize analog signals is to describe their frequency content. The frequency content of a signal is called the signal's **spectrum**.

For us, the spectrum of a signal means the combination of sinusoidal waves of different frequencies that make up a signal. For example, you're already somewhat familiar with the notion of a spectrum. If you shine a beam of white light on one side of a glass prism, multicolored light exits the opposite side of the prism as shown by the crude diagram in Figure 3-4. That's because light changes direction when it moves from one medium (air) to another medium (glass). That direction change is called *refraction*, and the amount of refraction depends on the frequency of the light. So a prism can be used to break light up into its constituent spectral colors. Figure 3-4 shows us that white light is, in fact, a combination of multiple colors of light.

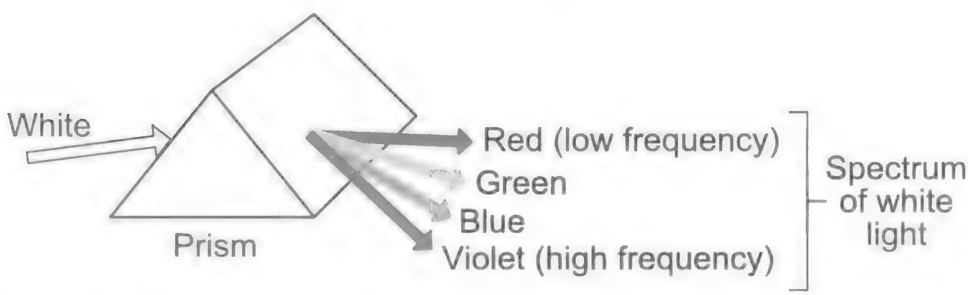


Figure 3-4 Breaking white light up into its constituent colors (spectrum).

By the Way

When you see a rainbow in the sky, water droplets in the atmosphere are behaving as individual prisms. The rainbow is caused by light being refracted while entering a droplet of water, then reflected from the back side of the droplet, and refracted again when leaving the droplet. If you're lucky enough to ever see a double rainbow, the second rainbow is caused by light reflecting twice inside water droplets. The neat thing about double rainbows is that the sequence of colors in the secondary rainbow is reversed from the order of the colors in the primary rainbow.

ANALOG SIGNAL SPECTRA

With this concept of spectrum in mind, engineers can actually display and measure an analog signal's spectrum using a spectrum analyzer like that shown in Figure 3-5. A **spectrum analyzer** displays the combination of sinusoidal waves of different frequencies that make up an analog signal. Let's look at a few examples of the spectra of analog voltage signals.

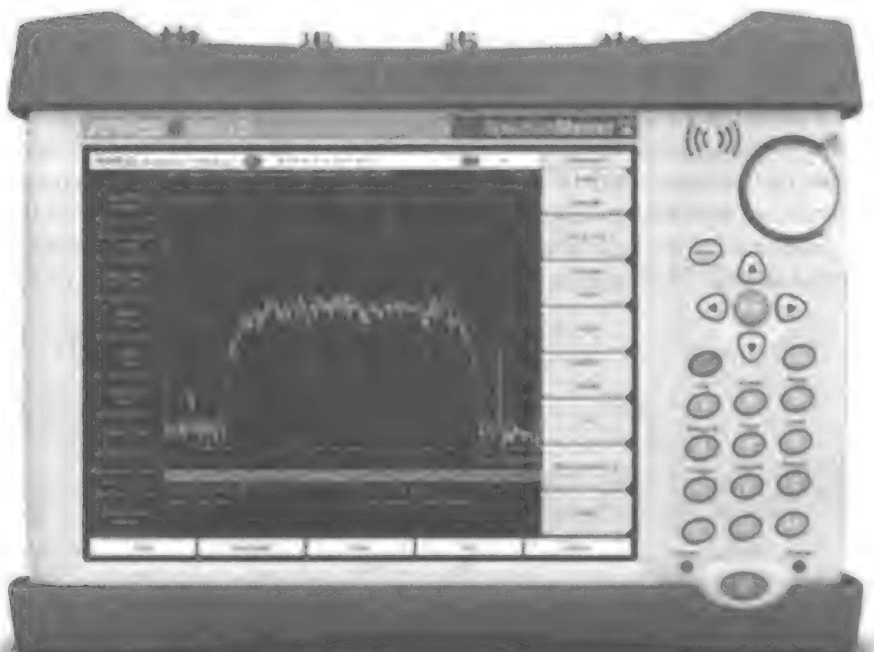


Figure 3-5 Commercial analog spectrum analyzer. (Courtesy of Anritsu Inc.)

Assume a microphone's output voltage is a 100 Hz sine wave as shown in Figure 3-6(a). Connecting that voltage to the input port of a spectrum analyzer would result in the analyzer's **frequency-domain** display, which is shown in Figure 3-6(b). The horizontal axis of frequency domain plots is always in units of frequency, typically Hz. The front panel controls of the analyzer are set so that the start frequency and the stop frequency of the analyzer's frequency display are 90 Hz and 110 Hz respectively.

The spectrum analyzer contains a tuned-frequency energy detector that is initially tuned to the start frequency of 90 Hz. At that frequency, the analyzer detects no energy at its input that oscillates at 90 Hz, so at the horizontal position of 90 Hz on its display screen, the analyzer shows no spectral energy. The tuned-frequency detector is then tuned to 91 Hz where it detects no input 91 Hz spectral energy and, again at the horizontal position of 91 Hz on its display screen, the analyzer shows no spectral energy. The same thing happens as the tuned detector is tuned from 92 Hz up to 99 Hz. Then, when tuned to 100 Hz, the energy detector indicates a high level of input energy that

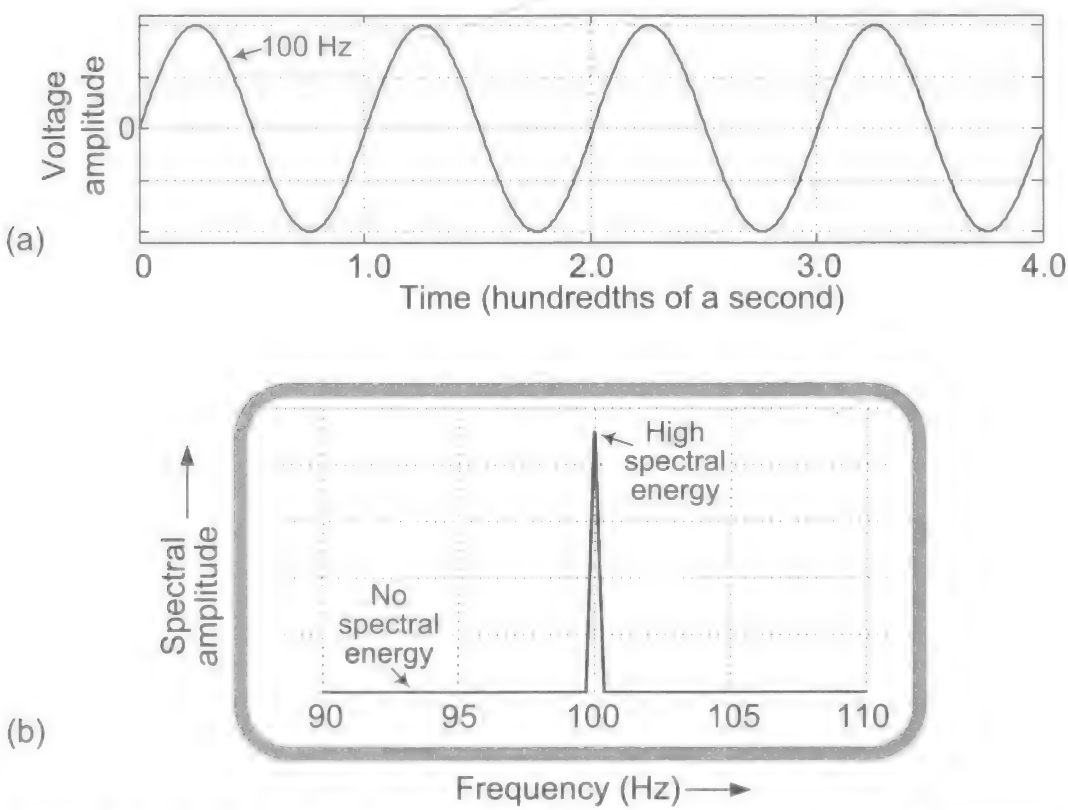


Figure 3-6 A 100 Hz analog sine wave: (a) time waveform; (b) spectrum analyzer spectral display.

oscillates at 100 Hz (100 cycles/second). That event causes the analyzer to display the high-level vertical spike at a frequency value of 100 Hz in Figure 3-6(b).

The analyzer's tuned-frequency detector is subsequently tuned, in turn, from 101 Hz to the stop frequency of 110 Hz where, in each case, no spectral energy is detected and the analyzer's display shows no input spectral energy in the frequency range of 101 to 110 Hz in Figure 3-6(b). That lone vertical spike in Figure 3-6(b) tells us that the voltage at the input of the spectrum analyzer contains a waveform oscillating at 100 Hz as we would expect by referring back to Figure 3-6(a).

Let's say an engineer wants to actually verify the frequency of the microphone's output sine wave voltage. There are two ways to do so. In the first method, the engineer could look at the voltage's time waveform using an oscilloscope, mentioned earlier, to see the display in Figure 3-6(a). Measuring the time duration of one oscillation to be one hundredth of a second, the engineer then knows that 100 oscillations occur over a time interval of one second. Thus, the sine wave's frequency is 100 Hz (100 cycles/second).

The second frequency measurement method is simpler. The engineer merely connects the sine wave voltage to the input of a spectrum analyzer, views the spectral display in Figure 3-6(b) to see the frequency location of the high-level spectral amplitude spike, and quickly determines that the sine wave's frequency is indeed 100 Hz.

The point is that engineers sometimes look at analog signals' amplitude waveforms over time using an oscilloscope, and sometimes they look at the spectral (frequency) content of analog signals, irrespective of time, using a spectrum analyzer. Spectrum analyzers are very powerful test instruments because their start and stop frequencies can be set to any values from tens of Hz to several GHz. An oscilloscope and a spectrum analyzer are as useful to engineers as a hammer and a saw are useful to carpenters.

With a spectrum analyzer, we could ping (vibrate) a tuning fork that has a tuned frequency unknown to us, near a microphone with its output cable connected to the input of a spectrum analyzer. Then, we could determine the tuned frequency of the tuning fork by observing the horizontal frequency location of the narrow spectral amplitude spike on the analyzer's display. Neat, huh?

A Composite-Signal Spectral Example

To strengthen our understanding of signal spectra, consider the high-amplitude 100 Hz sine wave represented by the dashed-line curve in Figure 3-7(a). The figure also has a lower-amplitude 200 Hz sine wave shown by the dotted-line curve. (Notice how the 200 Hz sine wave oscillates twice for each complete oscillation of the 100 Hz sine wave.)

The spectra of the 100 Hz and 200 Hz sine waves are shown in Figures 3-7(b) and 3-7(c), respectively. Because the amplitude of the 100 Hz wave is greater than the amplitude of the 200 Hz wave in Figure 3-7(a), the height of the spectral energy spike of the Figure 3-7(b) is greater than the height of the spectral energy spike in Figure 3-7(c).

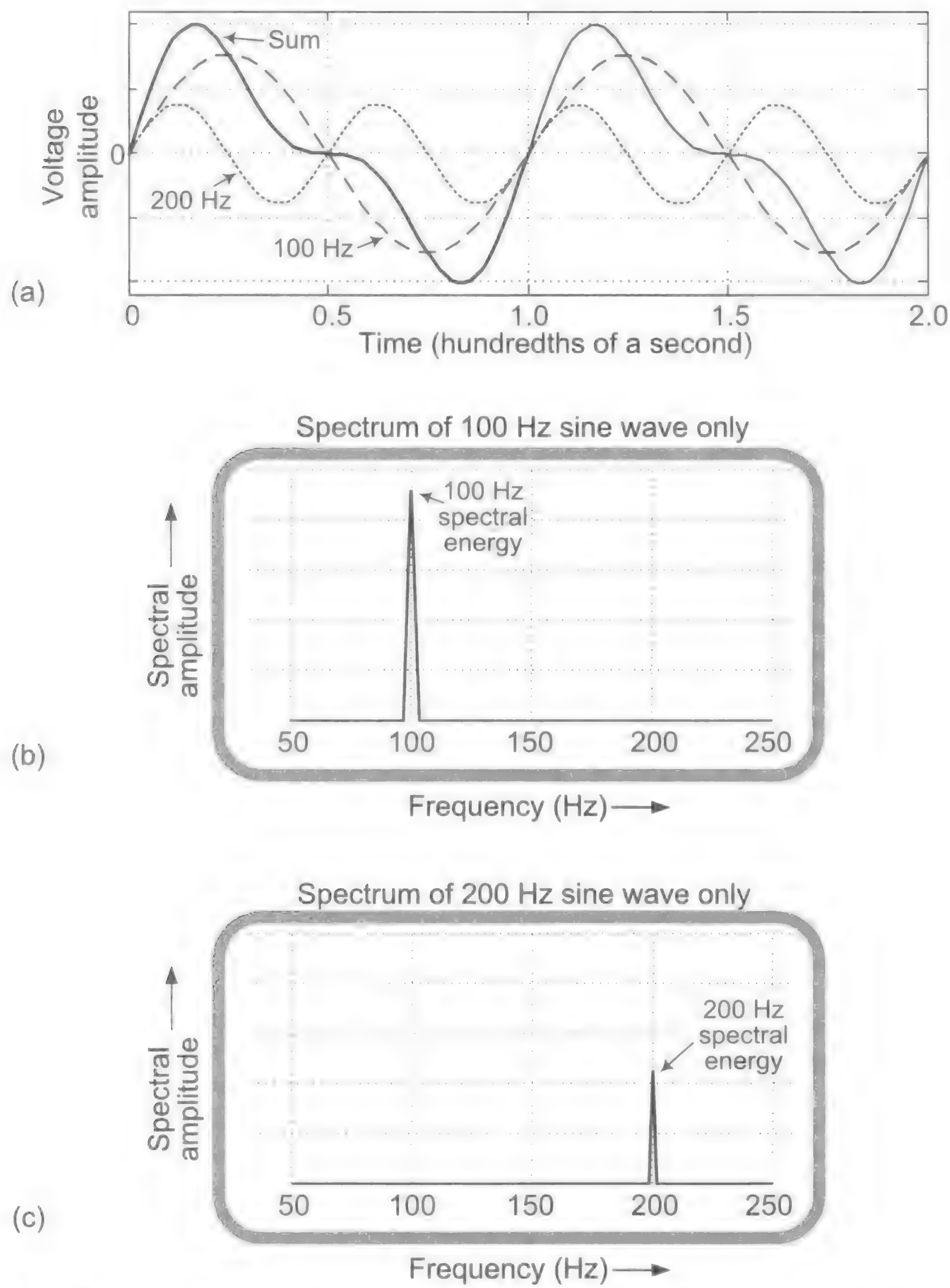


Figure 3-7 A 100 Hz sine wave and a 200 Hz sine wave: (a) time waveforms; (b) spectrum of the 100 Hz wave; (c) spectrum of the 200 Hz wave.

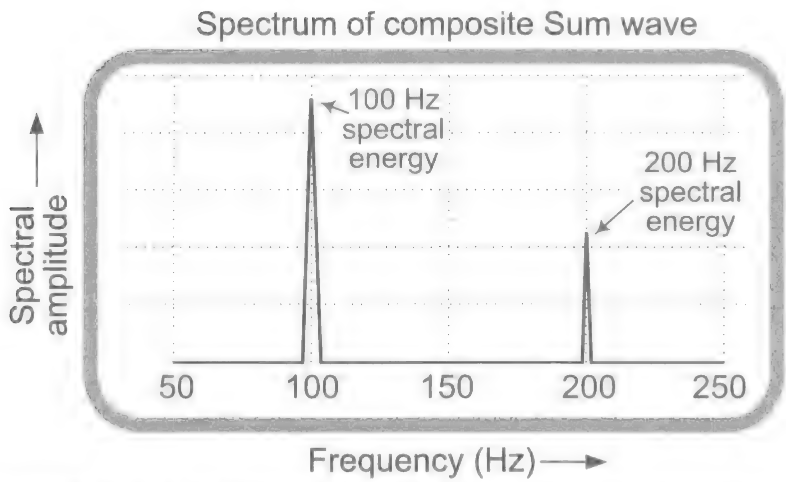


Figure 3-8 The spectrum of the composite Sum waveform.

If we added the 100 Hz sine wave and the 200 Hz sine wave, the result would be the composite Sum waveform shown as the solid-line curve in Figure 3-7(a). Notice that at any instant in time the solid-line Sum waveform is the sum of the dashed-line and dotted-line waves at that instant in time. For example, at the time instant of 1.25 hundredths of a second, the dotted 200 Hz curve equals zero, so the Sum curve is equal to the 100 Hz curve. At that instant in time, the Sum wave equals the 100 Hz wave plus zero.

The spectrum of the composite Sum wave in Figure 1-17(a) is shown in Figure 3-8. The spectrum in Figure 3-8 is the sum of the Figure 3-7(b) and Figure 3-7(c) spectra. From Figure 3-8 we learn one of the most important properties of analog signals:

> The spectrum of the sum of two analog time waves is equal to the sum of the waves' individual spectra.

Harmonics

There's an important topic we now introduce with regard to the spectrum of analog signals. That topic is **harmonics**, the undesirable spectral components in a time signal that cause an inadvertent distortion of the shape of the time signal. We start our discussion of harmonics by referring to the time-domain curves in Figure 3-9(a). In that figure, we see a high-amplitude 2 Hz sine wave represented by the solid-line curve. The figure also shows lower-amplitude 6 Hz and 10 Hz sine waves represented by the dashed- and dotted-line curves, respectively.

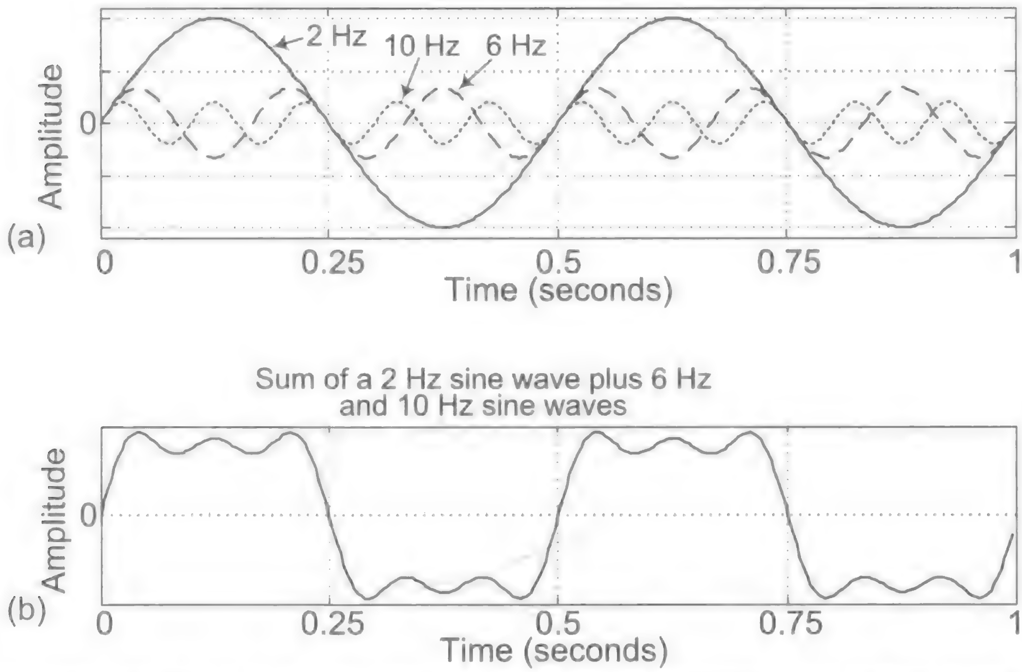


Figure 3-9 A 2 Hz analog square wave signal: (a) fundamental 2 Hz sine wave and its first two odd harmonics; (b) fundamental sine wave plus its first two odd harmonics.

If we add those three Figure 3-9(a) sinusoids together the sum is the waveform shown in Figure 3-9(b). If you'll forgive the corny analogy, we can say that if you add 1/4 cup of a 2 Hz sine wave, plus 4 teaspoons of a 6 Hz sine wave, plus 2.5 teaspoons of a 10 Hz sine wave, the combination would be the waveform in Figure 3-9(b). There, we see the Sum waveform looks much like a square wave. The 6 Hz and 10 Hz sine waves are called the "odd harmonics of the 2 Hz sine wave." That's because 6 and 10 are **integer** multiples of 2; that is, $6 = 3 \times 2$ and $10 = 5 \times 2$. The numbers 3 and 5, both of which are odd numbers, are **integers** (whole numbers). We're free to call the Figure 3-9(b) waveform a "2 Hz square wave" because it repeats itself twice in the time interval of one second.

In addition, adding the low-amplitude levels of 14 Hz (7×2) and 18 Hz (9×2) harmonic sine waves, shown in Figure 3-10(a), to the sum of sine waves in Figure 3-9(b) results in the even more square-like square wave we see in Figure 3-10(b).

You can see what's happening here. The more of its odd harmonics we add to a 2 Hz sine wave, the closer the summation waveform looks like a 2 Hz square wave. And that's why we can say that a square wave is made up of, or comprises, a fundamental frequency sine wave plus that sine wave's odd harmonics. There's nothing

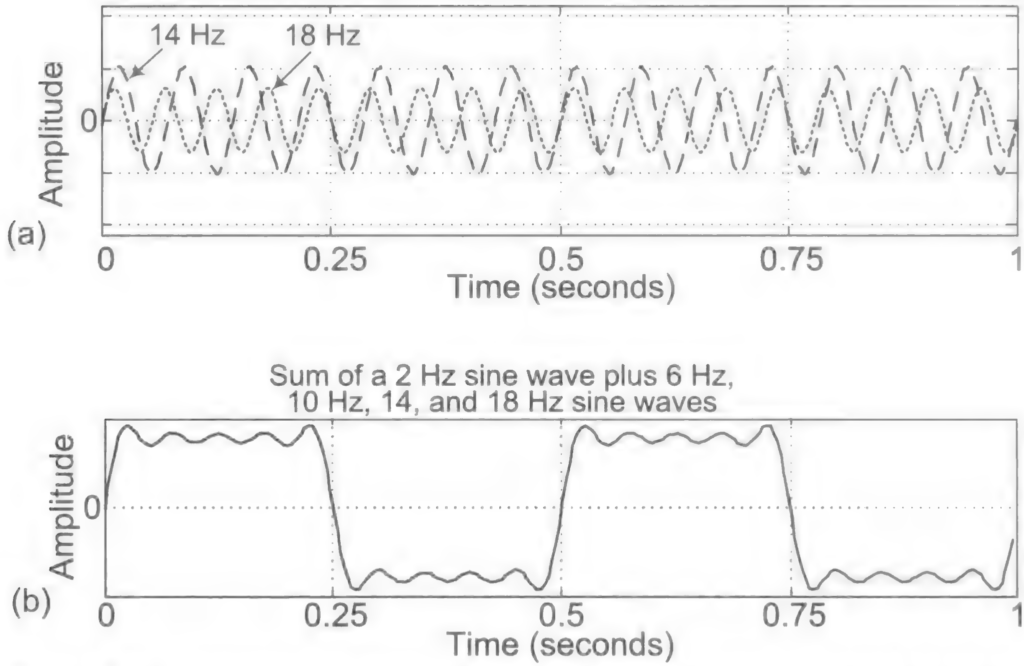


Figure 3-10 *A 2 Hz square wave signal: (a) fundamental 14 Hz and 18 Hz sine waves; (b) fundamental 2 Hz sine wave plus its first four odd harmonics.*

sacred about starting with a 2 Hz sine wave here. We could just as well add a 5 kHz (5,000 Hz) sine wave to its odd harmonic sine waves of 15 kHz, 25 kHz, 35 kHz, and 45 kHz and obtain the square waveform in Figure 3-10(b) that repeats itself 5,000 times in one second.

The spectrum of the Figure 3-10(b) waveform is shown in Figure 3-11. The 2 Hz spectral component in Figure 3-10 is called the "fundamental frequency" because the Figure 3-10(b) waveform's repetition rate is two repetitions per second (two cycles/second).

Your authors realize that spectral plots like Figure 3-11 may be new and a bit puzzling to many readers. Signal processing engineers interpret that figure as follows: Figure 3-11 simply tells us that a time waveform, shown in Figure 3-10(b), contains some given amplitude of a 2 Hz sine wave, plus a lower amplitude 6 Hz sine wave, plus lower-amplitude 10 Hz, 14 Hz, and 18 Hz sine waves.

To illustrate that a square wave really does contain odd harmonics, let's say we built an audio detection system, containing a microphone carefully designed to detect the presence of an 18 Hz sine wave audio tone. When an 18 Hz audio tone is detected, the system turns on a red warning light. Now, if we connected the 2 Hz square wave voltage shown in Figure 3-10(b) to the terminals of a loudspeaker, the 18 Hz detection

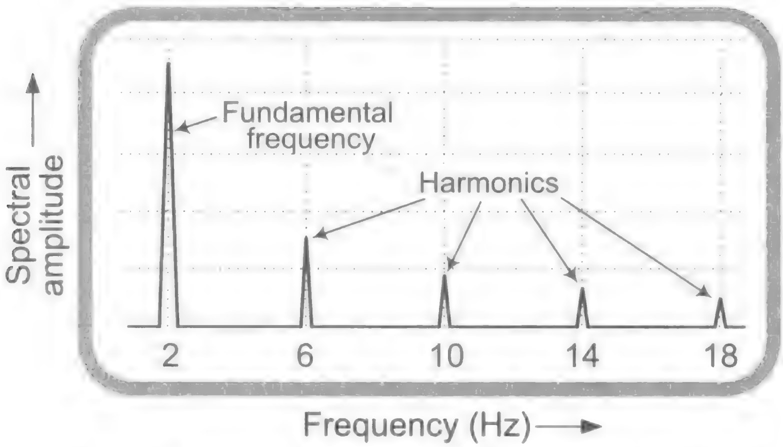


Figure 3-11 Spectral display of a 2 Hz analog square wave.

system's warning light would suddenly light up. The audio tone detection system would identify the 2 Hz audio square wave's 18 Hz harmonic component that we showed in Figure 3-11. Single-frequency sine wave voltages contain no harmonics but square wave voltages certainly do.

One consequence of this harmonics discussion is that we learn one of the most important principles in all of signal processing. That is:

> A time signal having very abrupt (sudden) amplitude changes, like a square wave, contains higher-frequency spectral content than a single-frequency sinusoidal signal that has more gradual (smooth) amplitude changes.

This characteristic must be anticipated during the design of all analog and digital signal processing systems.

Harmonic Distortion

In practice, harmonics can be detrimental. Here's why: think of the nice clean analog 5 Hz sine wave shown in Figure 3-12(a) with its frequency-domain spectrum that is shown in Figure 3-12(b).

Now let's say the 5 Hz sine wave voltage in Figure 3-12(a) was applied to an amplifier whose operation was flawed, and the output of the amplifier is the distorted voltage waveform shown in Figure 3-13(a). The amplifier's imperfect performance flattened the positive and negative peaks of the original pure sine wave input signal. That distortion is called **harmonic distortion** because the spectrum of the amplifier's

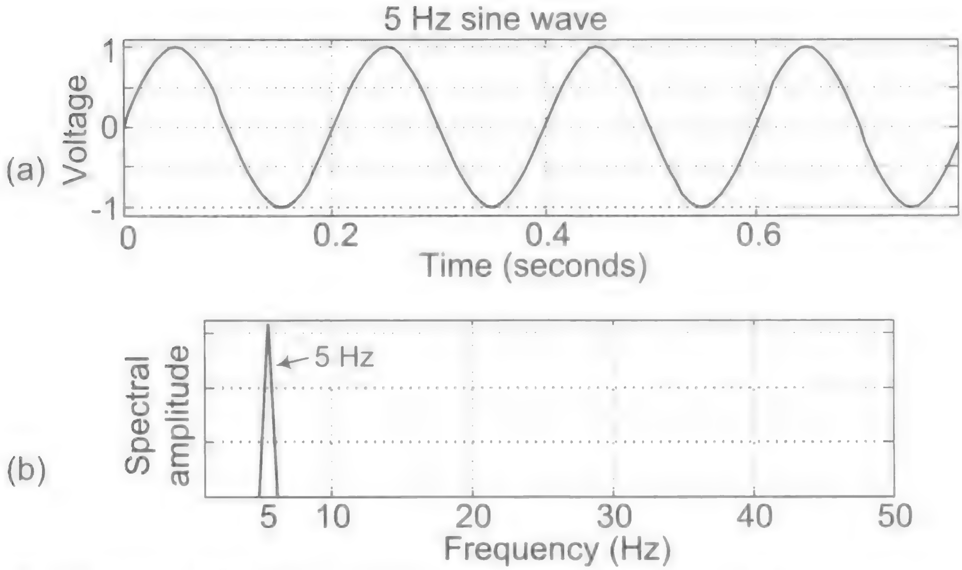


Figure 3-12 A 5 Hz sine wave voltage: (a) time-domain plot; (b) spectral plot.

output signal contains unwanted harmonic spectral components as depicted in Figure 3-13(b). Let's be clear now: The harmonics in Figure 3-13(b) did not *cause* the distorted waveform in Figure 3-13(a). The fact that the Figure 3-13(a) waveform is distorted is what produced the harmonics in Figure 3-13(b).

As an engineering example of harmonic distortion, if we replaced the analog sine wave in Figure 3-12(a) with the radio signal produced by a broadcast radio station, the distorted output of an imperfect amplifier would be the desired radio station signal plus copies of that signal centered at higher frequencies. That scenario is strictly forbidden. We don't want harmonics from one radio station interfering with the signal from another station in a nearby city. That's why the transmitters of broadcast radio stations are carefully designed to radiate signals only at their Federal Communications Commission–specified frequencies. Likewise, your cell phone contains carefully designed filters so that it does not radiate undesirable, high-frequency, harmonic electromagnetic energy. We don't want your phone's transmitted signal to interfere with someone else's phone transmissions.

However, with audio signals, harmonics can be a good thing. Musical instruments generate an audio tone plus distinctive multiple audio harmonics. These harmonics allow us to tell musical instruments apart. Without harmonics, a piano playing a middle-C note would sound the same as a guitar playing middle C.

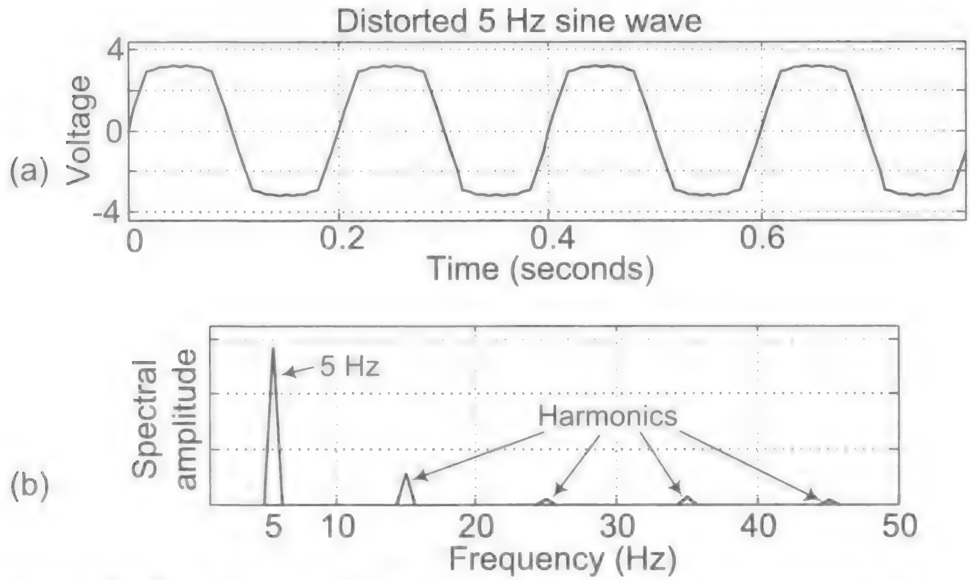


Figure 3-13 *A distorted 5 Hz sine wave voltage: (a) time-domain plot; (b) frequency-domain spectrum showing harmonic distortion.*

By the Way

Decades ago, rock-n-roll guitarists realized that if they applied unusually high-level electric guitar signals to their vacuum-tube amplifiers, shortcomings in those amplifiers would produce musical notes suffering from high harmonic distortion such as that in Figure 3-13(b). The result was musical notes super-rich in roaring harmonic content, a sound that the rock guitarists liked. When transistor amplifiers came along, with their improved performance, the harmonic distortion was reduced so some rock guitarists weren't happy with the first transistor amplifiers. This made the old vacuum-tube amplifiers highly sought after. Rumor has it that Rolling Stones lead guitarist Keith Richards still prefers vacuum-tube amplifiers. Because Richards has not returned our phone calls, your authors cannot confirm this rumor.

Bandwidth

Now that we've covered the basics of analog signal spectra, we can proceed to another important topic, signal **bandwidth**, which is the frequency range over which a signal contains significant spectral energy. A good way to start our bandwidth discussion is by considering old landline telephone systems.

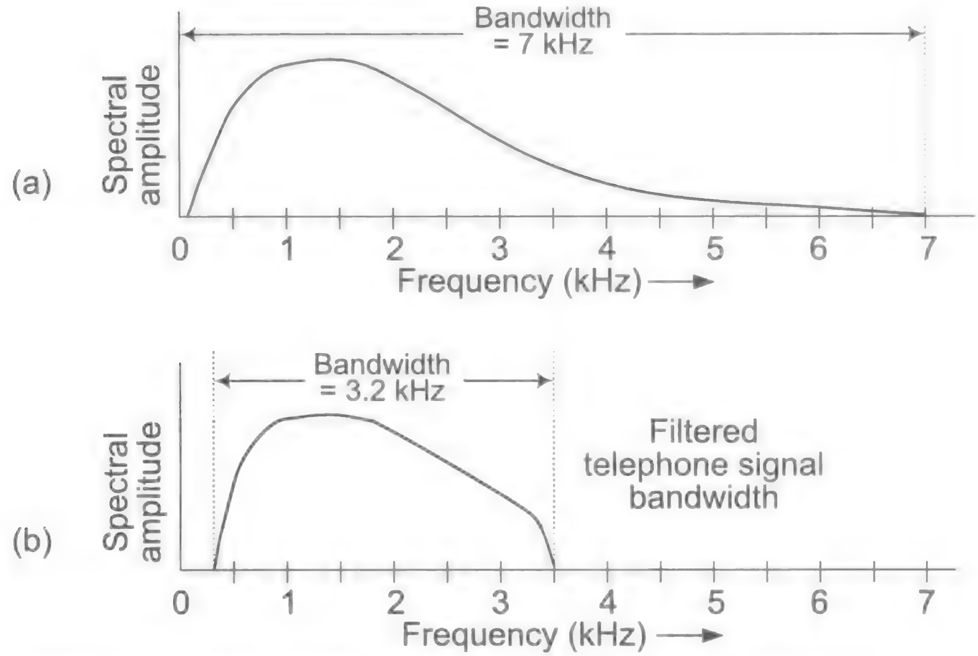


Figure 3-14 Human speech bandwidth: (a) full spectral bandwidth; (b) spectral bandwidth after telephone company filtering.

The audio spectrum of human speech looks something like the curve shown in Figure 3-14(a). When people speak into the microphone of a landline telephone, their speech signals contain spectral energy that covers a frequency range of roughly 80 Hz to 7 kHz (7,000 Hz), so it's reasonable to say that human speech has a bandwidth of 7 kHz.

For a number of practical engineering reasons, the telephone company requires that an analog telephone speech signal contain energy only over a frequency range of somewhat less than 4 kHz. As a result, at the telephone company's facility a telephone speech signal is passed through an analog filter that effectively eliminates spectral energy below 300 Hz and above about 3.5 kHz. Thus, the audio signal that the telephone company transmits from one telephone to another is limited in its bandwidth to 3.2 kHz, as shown in Figure 3-14(b).

There are a number of different definitions for bandwidth used by engineers. However, for now we'll consider the word *bandwidth* to mean the frequency range over which a signal contains significant spectral energy. The reason the telephone company could limit the bandwidth of analog speech signals to 3.2 kHz is because most of the spectral energy of human speech is in the range of 300 Hz to 3.5 kHz, and humans can easily understand a speech signal that is limited in bandwidth to 3.2 kHz. It's not high-fidelity speech, but certainly good enough to hold a conversation over the telephone.

Amplitude modulation (AM) radio stations broadcast audio signals that have bandwidths of 5 kHz. That's why the audio from your AM radio sounds better than

telephone audio. Better still, **frequency modulation (FM)** radio stations broadcast audio signals, for each of the left and the right channels, that have bandwidths just slightly less than 15 kHz. That's why orchestra music sounds so good coming from an FM radio station. Audio geeks refer to FM audio as **high fidelity**. (Appendix C provides additional information concerning AM and FM radio signals.)

Let's conclude our analog-signal bandwidth discussion by returning to an audio signal we discussed earlier. Figure 3-15(a) shows the spectrum of Capt. Kirk's "Mister Spock" audio speech signal in Figure 2-12(a). Note that Figure 2-12(a) is a time-domain plot and Figure 2-15(a) is a frequency-domain plot. In Figure 3-15(a), we see that the speech signal has a bandwidth of roughly 4 kHz.

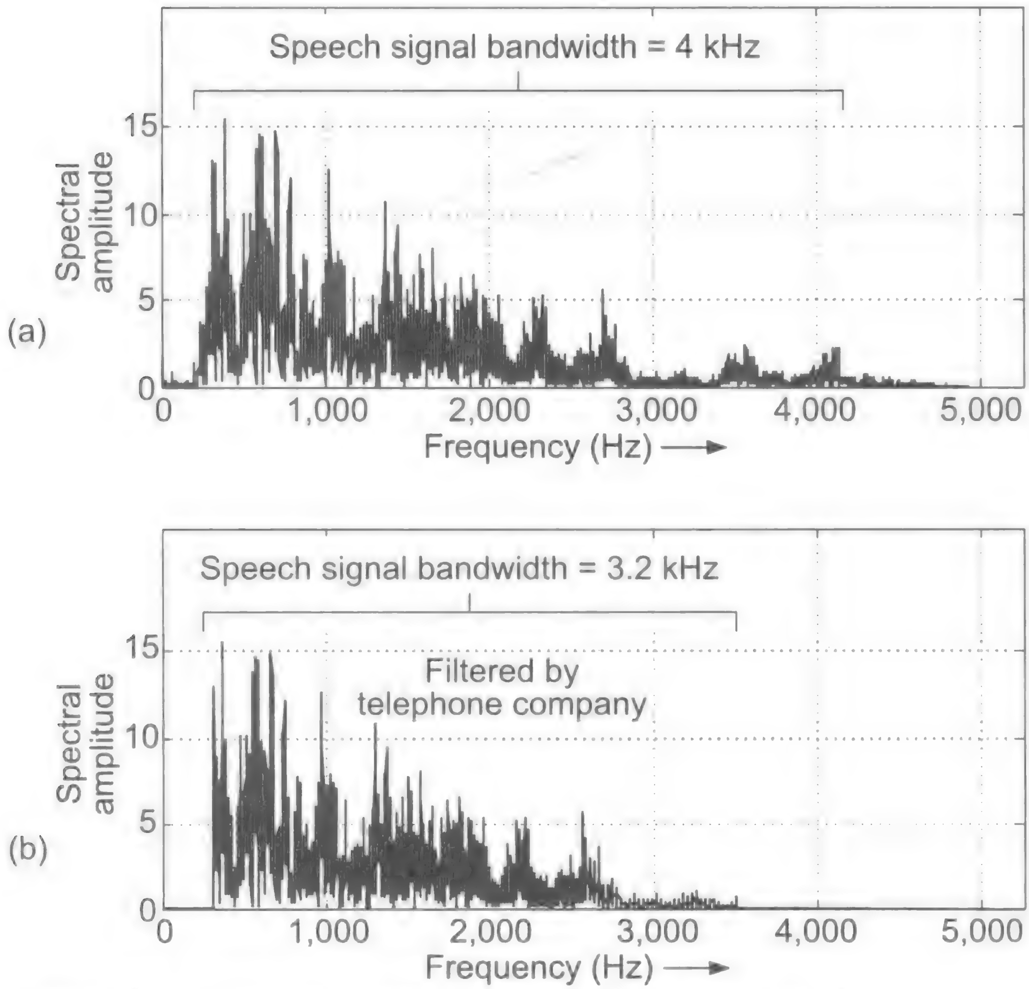


Figure 3-15 Spectrum of the audio speech signal "Mister Spock."

If that audio speech voltage signal had been passed through the telephone company's filter, which only passes spectral energy in the range of 300 Hz to 3,500 Hz, the output spectrum of the filter would be that shown in Figure 3-15(b). Do you see the difference between the spectra in Figure 3-15(a) and Figure 3-15(b)? There you go—you now have experience in examining signal spectra! Because the vast majority of the spectral energy in Figure 3-15(a) does pass through the filter, the filtered speech signal would be easily perceptible at the receiving telephone's loudspeaker output.

The Other Bandwidth

Unfortunately, there is an incorrect definition of the word *bandwidth* that's *very* common nowadays. When you buy a new cell phone, the salesperson may tell you something like, "Your phone's bandwidth is 1.4 megabits/second." What he or she should have said was, "Your phone's *binary data transfer rate* is 1.4 megabits/second." (We explain binary bits in a later chapter.) For us, the term *bandwidth* has the very specific meaning of *the width of a frequency band*, measured in Hz, *not* a binary data transfer rate.

We don't know just when the word *bandwidth* began to be used to describe binary data transfer rates but, sadly, that misnomer is surely here to stay.

WHAT YOU SHOULD REMEMBER ....

In this chapter, we covered the concept of frequency as well as how we can describe an analog signal by its frequency-domain spectrum. All of the concepts we learned about the spectra of analog signals will help us understand digital signals.

The concepts you should remember from this chapter are:

- Periodically varying voltages, like sine and cosine wave voltages, are common in the world of signal processing.
- The repetition rate, per unit time, of a periodic voltage is called frequency.
- Frequency is most commonly measured in units of hertz (Hz). One Hz is equal to one cycle per second.
- We can describe analog signals by their frequency-domain spectral content.
- The spectrum of periodically varying voltages, except single-frequency sine and cosine wave voltages, contains a fundamental frequency component plus harmonic frequency components (higher frequency sinusoids).
- The frequency range over which a signal contains significant spectral energy is called the bandwidth of the signal.

4 Digital Signals and How They Are Generated

WHAT IS A DIGITAL SIGNAL?

Until the early 1980s, the vast majority of the signals that we experienced in our daily lives, such as light and sound signals, were analog in nature. But the advent of digital clocks, cell phones, portable music players, and digital television has changed much of our world to digital as indicated by Table 1.1 in Chapter 1. So it's reasonable to ask, "What does the word *digital* mean? What is a **digital signal**?" We answer those questions in the following sections.

The Notion of Digital

The development of language is a complicated, and unpredictable, phenomenon. As far as we can tell, our definition of the word *digital* originated from the notion of counting with your fingers. (The Latin word *digitus* means finger.) If someone asked you to hold up any nonzero number of fingers from your right hand, you have five choices. You could hold up 1, 2, 3, 4, or 5 fingers (yes, the thumb is promoted to a finger for this discussion). The number of upheld fingers thus will be one of only five possibilities. As such, we could say that the number of fingers you hold up is a digital number. So, for us the word *digital* means one possibility selected from a well-defined number of discrete possibilities. A **digital number** is a single number selected from a fixed set of numbers. Don't worry; we'll clarify the meaning of digital with many examples.

As for the term *digital signal*, unfortunately there are two accepted definitions in two different fields of electronics. Those two definitions are quite distinct so we'll carefully describe both definitions in the following sections.

Digital Signals: Definition #1

The first definition of a digital signal is actually related to what we've been calling an **analog signal** in this book. There are analog voltage signals in cell phones, high-definition televisions, and computers that look like the signal shown in Figure 4-1. (Engineers can actually see the voltage waveforms shown in Figure 4-1 with an oscilloscope such as that depicted in Figure 2-3(a).)

Those voltages, inside electronic hardware, represent some sort of information. For example, when you print a document on your desktop printer, voltages like that shown in Figure 4-1 are transmitted from your computer through a cable to your printer. Your printer is designed to interpret the voltage signal and print the appropriate letters, numbers, or perhaps a two-dimensional image (a picture). The Figure 4-1 voltage waveform is often referred to as a **digital signal** because the voltage at any instant in time is at one of only two possible voltage levels. At any instant in time, the voltage in our example is either zero volts or 4 volts.

The thoughtful reader would now ask, "Wait a second! If I drew the Figure 4-1 voltage on a piece of paper, the tip of my pencil would never leave the surface of the paper. Isn't that a primary characteristic of *analog* signals?" The answer is yes. The voltage in Figure 4-1 is indeed an analog voltage signal, but because its voltage value at any instant in time is always one of two possible values, it's called a digital signal by electronic hardware design engineers. So for our purposes, we'll merely accept the fact that the Figure 4-1 voltage is commonly called a digital signal by engineers who build electronic hardware.

> Digital signal definition #1: An analog voltage signal that alternates between two distinct voltage values.

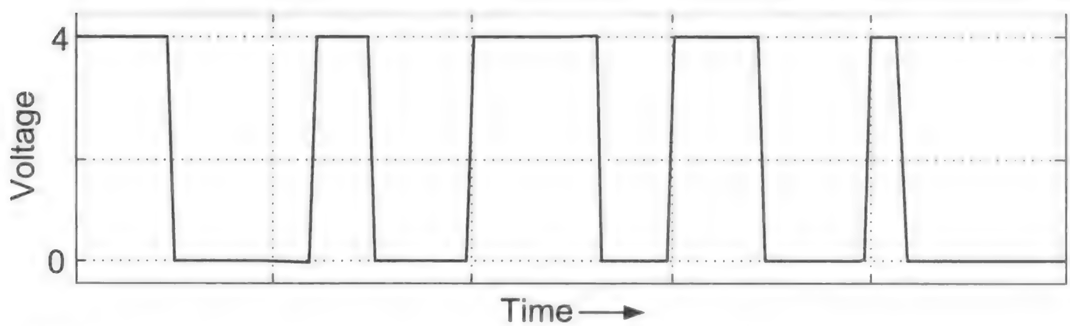


Figure 4-1 *An analog voltage signal that's called a digital signal.*

Digital Signals: Definition #2

The second and more common definition of digital signal requires careful explanation. As an everyday example, when you look at a standard wall clock with its rotating minute and hour hands, the minute hand may point to the number 2 telling us it's 10 minutes past the hour, as shown in Figure 4-2(a). Over the next 60 seconds, the minute hand slowly rotates to 11 minutes past the hour as shown in Figure 4-2(b). On a digital clock, however, at 10 minutes past the hour the minutes display will show the value 10. And for the next 59 seconds, the minutes display continues to show the number 10. Then all of a sudden, the digital clock's minute display quickly switches to the number 11. Unlike a rotary-hand clock, a digital clock cannot show any time between 10 and 11 minutes past the hour. So, the minutes display on a digital clock is a digital number that can only be one of 60 possible whole numbers, either 00, 01, 02, 03, . . . , 58, or 59.

In a similar example, the typical automobile dashboard speedometer shows the car's speed, in miles per hour (mph), as depicted in Figure 4-3(a). The speedometer's pointer moves smoothly from left to right. However, some modern autos have digital speedometers that display an easy-to-read number, as shown in Figure 4-3(b). We are at liberty to call that number 48 in Figure 4-3(b) a digital number, as we did with the minutes display on a digital clock.

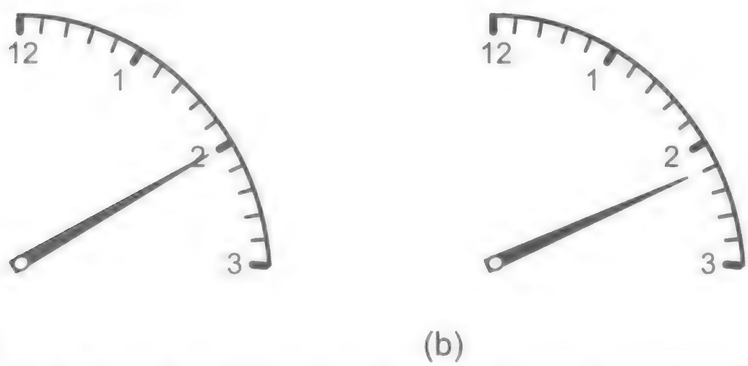


Figure 4-2 A rotary clock's minute hand positions: (a) 10 minutes after the hour; (b) 11 minutes after the hour.

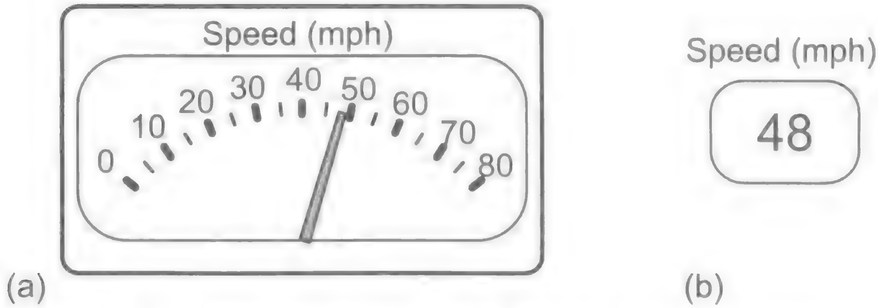


Figure 4-3 Automobile speedometer displays: (a) old-style analog pointer display; (b) digital display.

OK, let's now consider a sequence of digital numbers. We start by recalling the low and high Marquette, Michigan, temperature curves in Figure 2–1. We repeat the low temperature curve here in Figure 4-4.

Instead of having the Figure 4-4 low temperature analog curve available to us, assume we have the following table of low temperatures obtained on the first day of each month over a period of one year.

If we plot Table 4.1's sequence of 12 numbers as dots, one dot per number, we have the plot shown in Figure 4-5. That plot of dots is important because the list of temperature values in Table 4.1 is also called a digital signal and the plot in Figure 4-5 is a graphical depiction of this signal.

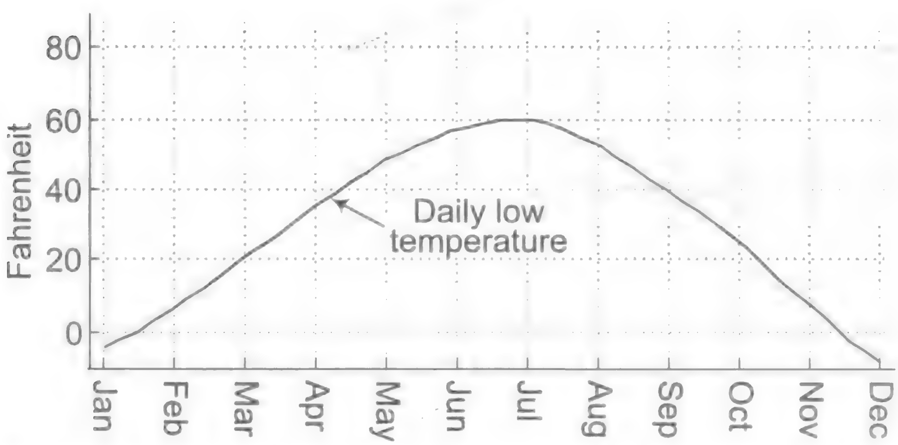


Figure 4-4 Average low outdoor temperatures in Marquette, Michigan.

Table 4.1 Low Temperatures (First Day of the Month)

Month	Low Temp.	Month	Low Temp.
January	−3°	July	59°
February	6°	August	51°
March	19°	September	39°
April	33°	October	22°
May	46°	November	6°
June	55°	December	−2°

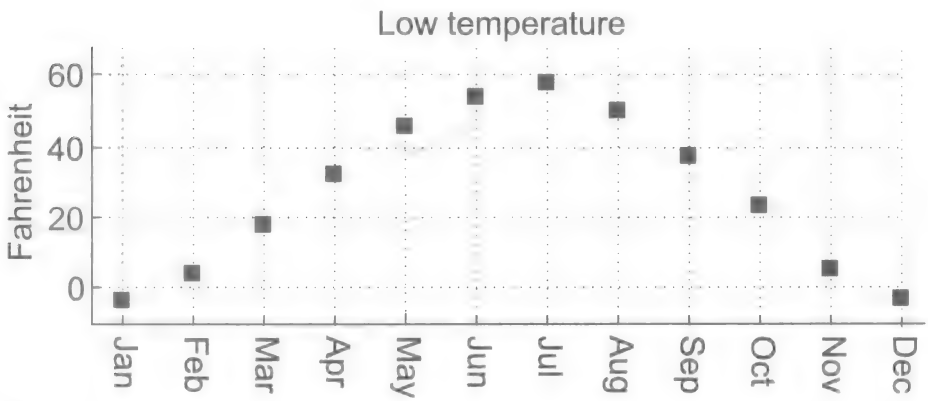


Figure 4-5 Discrete sequence of low outdoor temperatures.

To establish our terminology here, we say the list of 12 numbers in Table 4.1 can be called:

- a digital signal (the least appropriate but by far the most common terminology),
- a *digital sequence* (occasionally used), or
- a *discrete sequence* (the most appropriate terminology but not commonly used).

We will comply with common terminology and state:

> Digital signal definition #2: A sequence of discrete, individual numbers. This is the definition we'll use throughout the remainder of this book.

We turn now to the fundamental difference between the smoothly changing analog temperature signal in Figure 4-4 and the discrete, periodically spaced-in-time,

digital temperature signal in Figure 4-5. Both signals tell us how the low outdoor temperature varies over a period of one year in Marquette, Michigan. Later, we'll learn in what situations digital signals are more useful to us than are analog signals.

HOW DIGITAL SIGNALS ARE GENERATED...............

There are three common methods used to generate digital signals. We discuss each method in the following paragraphs.

Digital Signal Generation by Observation

The digital signal in Figure 4-5 was obtained by observation. Over a year's time, on the first day of the month, someone observed a thermometer and recorded the coldest temperature for that day. The observer then listed those 12 temperatures in Table 4.1. The digital signal, representing the physical quantity of temperature listed in Table 4.1, was obtained by observation.

Another example of a digital signal generated by observation is the daily closing prices of one share of Apple (Apple Inc.) stock. The prices, observed over a three-year period and measured in U.S. dollars, form a digital signal, a discrete sequence of numbers, as shown in Figure 4-6(a).

Stock market analysts routinely record, compile, and examine historical daily stock prices like these. However, rather than plotting a sequence of prices as dots, similar to what we did in Figure 4-6(a), analysts connect the dots with lines and then delete the dots, producing a jagged curve as shown in Figure 4-6(b). Note that the curve in Figure 4-6(b) is *not* a continuous analog signal. It's merely the popular way to graphically depict the digital data in Figure 4-6(a).

As for the information content of Figure 4-6, many readers will find that figure of little interest. However, for those readers who bought or sold shares of Apple Inc. stock during the summer of 2012, Figure 4-6 contains crucial information.

By the Way

Thanks to the Internet, you can examine the historical stock share prices of any company registered with any major stock exchange. You may want to visit this Web site, http://finance.yahoo.com, to check the historical and current stock price of your employer.

Digital Signal Generation by Software

Another way digital signals are generated is through the use of computer software. Engineers frequently use software to create digital signals needed for analysis or testing

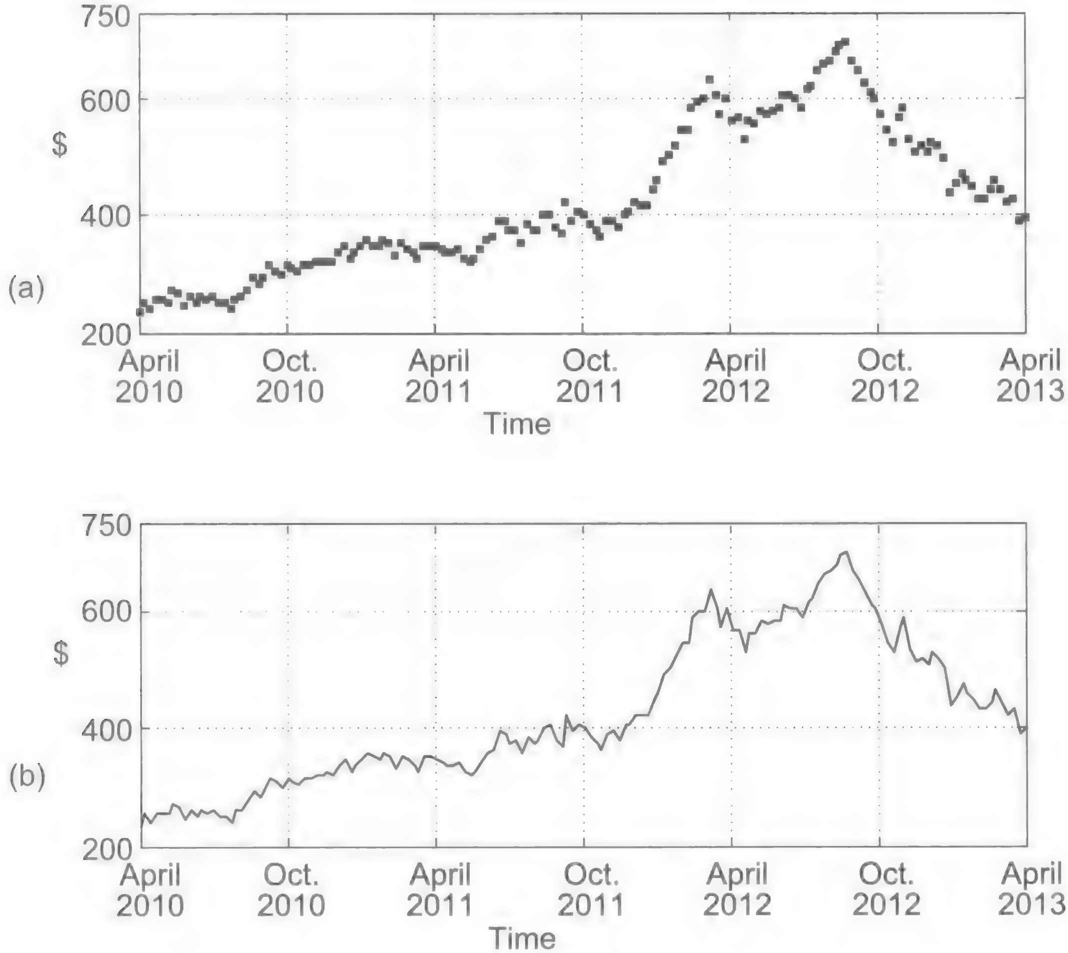


Figure 4-6 Closing price of one share of Apple stock: (a) daily prices plotted as dots; (b) preferred display method of connecting the dots with lines, and then deleting the dots.

purposes. The results of their efforts are digital signals, lists of numbers stored in a computer's memory that may represent any physical quantity of an engineer's choice. In later chapters, we'll look at several examples of software-generated digital signals.

Digital Signal Generation by Sampling an Analog Signal

By far the most common way digital signals are generated is by **sampling**, a process by which we represent an analog signal with a list of numbers. The process of

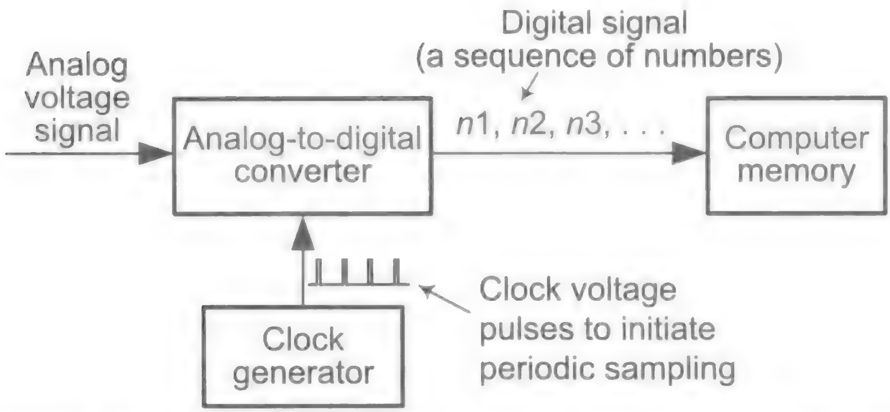


Figure 4-7 *Sampling: converting an analog signal to a digital signal for storage in computer memory.*

sampling is depicted in Figure 4-7. An analog voltage signal is applied to a small electronic device called an **analog-to-digital converter**. For convenience, we'll refer to an analog-to-digital converter as an **ADC**. The output of the ADC is a sequence of numbers, $n1$, $n2$, $n3$, . . . , that is stored in a computer's memory. The sequence of numbers is our digital signal obtained by sampling an analog voltage signal.

Critical to our sampling process are the periodically spaced *clock* voltage pulses applied to the ADC. Those pulses determine the exact instants in time when the converter will measure the analog voltage's instantaneous value and generate a single output number representing that value. Let's look at an example of sampling an analog voltage signal.

We show the input and output of a sampling process in Figure 4-8. Let's assume the analog voltage input signal to an ADC is the sinusoidal voltage shown in Figure 4-8(a). As shown in that figure, we sample the analog ADC input voltage at the equally spaced instants in time represented by the vertical arrows.

The sampled values, the sequence of 20 numbers output by the ADC, are listed in Table 4.2. Those sample values are also graphically depicted as dots in Figure 4-8(b). Each dot represents one sample value, one number, of the sampled sequence.

If we took it upon ourselves to draw the ADC output sequence on a piece of paper with time on the horizontal axis as in Figure 4-8(b), the tip of our pen or pencil would indeed leave the surface of the paper, unlike drawing a continuous analog signal.

The sequence of numbers in Table 4.2 comprises the ADC's output digital signal, and those numbers can be stored in a computer's memory. We refer to the ADC output sample values, the sequence numbers represented as dots in Figure 4-8(b), as a *sampled version* of the analog signal in Figure 4-8(a). So there you are. The sequence of numbers listed in Table 4.2, and represented by the dots in Figure 4-8(b), is a digital signal.

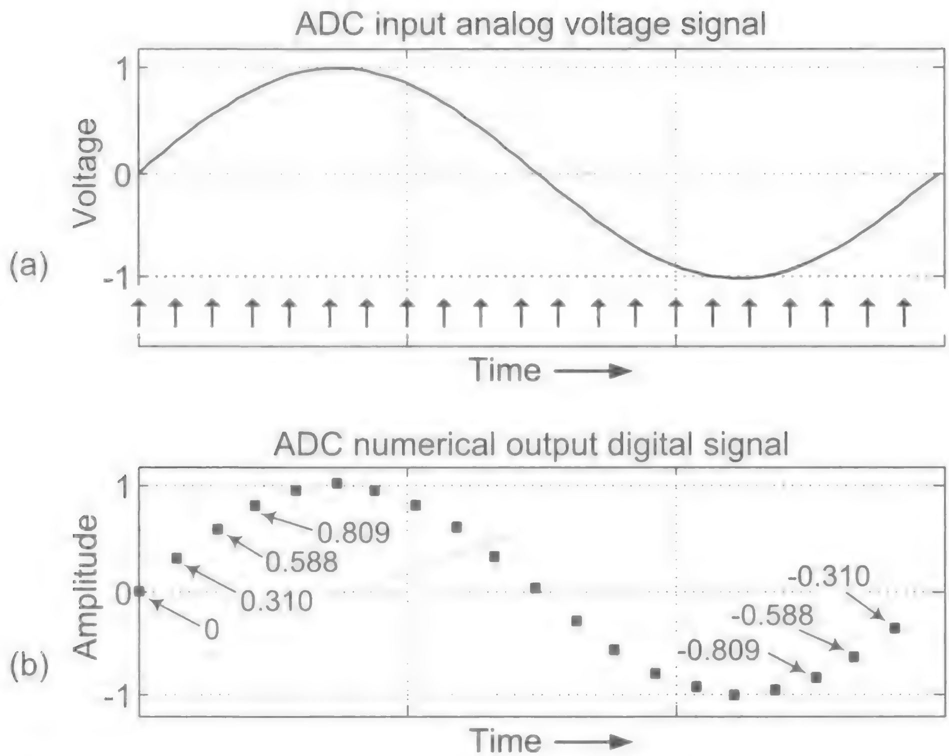


Figure 4-8 *Analog-to-digital converter operation: (a) analog signal input; (b) digital signal output.*

Table 4.2 Sample Values of a Sinusoidal Digital Signal

1st sample = 0	11th sample = −0.001
2nd sample = 0.310	12th sample = −0.309
3rd sample = 0.588	13th sample = −0.588
4th sample = 0.809	14th sample = −0.808
5th sample = 0.951	15th sample = −0.951
6th sample = 1.0	16th sample = −1.0
7th sample = 0.951	17th sample = −0.951
8th sample = 0.808	18th sample = −0.809
9th sample = 0.588	19th sample = −0.588
10th sample = 0.309	20th sample = −0.310

The Sample Rate of a Digital Signal

In the world of digital signal processing (DSP), every digital signal sequence of numbers has what is called a **sample rate** associated with it. The **sample rate** of a digital signal is the repetition rate of the signal's samples, measured in samples per unit of time. And the sample rate of a digital signal is extraordinarily important. In Figure 4-7, the sample rate of the $n1$, $n2$, $n3$, . . . digital signal is the repetition rate (the frequency) of the clock signal used to initiate the conversion of the analog voltage signal. The notion of sample rate, how often an analog signal is sampled, is fairly easy to understand.

For example, if someone sent you an e-mail listing the 20 digital signal sample values in Table 4.2 and you created your own version of Figure 4-8(b), you're missing an important piece of information. You would have no idea what the frequency is of the analog sine wave that was sampled to produce the digital signal in your drawing. Did the sine wave repeat every second, every 10 seconds, or once a week? On the other hand, if you knew what the sample rate of the e-mailed digital signal was, say 60 samples/second, you would then be able to step through the arithmetic and determine that:

- 60 samples spans 1 second of time,
- 20 samples spans 1 cycle of the analog sine wave,
- 20 samples is one-third of a second (20/60 = 1/3),
- 1 cycle of the analog signal lasts one-third of a second,
- 3 cycles of the analog sine wave last 1 second, and
- the frequency of the analog sine wave is 3 cycles/second, or 3 Hz.

Again, to fully understand the digital signal represented by the dots in Figure 4-8(b), we need to know the sample rate of that sequence of numbers.

To test your understanding of sample rate, we ask: what are the sample rates of the digital signals in Figure 4-5 and in Figure 4-6(a)? Hopefully, your answer is: the sample rate for the digital signal in Figure 4-5 is one sample per month and the sample rate for the digital signal in Figure 4-6(a) is one sample per day.

A SPEECH DIGITAL SIGNAL

Let's now look at a slightly more complicated digital signal. In Chapter 2, we discussed an analog voltage representing the audio speech signal of Capt. Kirk speaking the words "Mister Spock." We also looked, in greater detail, at the analog audio signal of the first syllable, "Mis," of the word *Mister*. Figure 2-12(b) shows the details of the analog voltage waveform of the audio syllable "Mis."

The sampled version of that analog audio "Mis" signal, a digital signal, is shown by the dots in Figure 4-9(a). As a digital signal, the syllable is more complicated than we would expect, as we see in Figure 4-9(a), where it looks like a dense jumble of dots. To help us understand the nature of the signal, we could connect the dots with straight lines as in Figure 4-9(b). However, this is only a slight improvement visually. For more visual clarity, DSP folks connect the dots in Figure 4-9(a) with lines and then delete the dots to obtain Figure 4-9(c). This simplified version shows the fluctuating amplitude of the digital signal so that it's easier to visualize and understand. Yes, Figure 4-9(c) looks like an analog signal. But in looking at this sort of figure, DSP engineers realize that the figure represents a digital signal (a sequence of discrete numbers) rather than a continuous analog signal.

The sample rate of the Figure 4-9 digital speech signal is 11,025 samples/second. That sample rate might seem like a strange value but it's a sample rate commonly used by audio signal processing engineers. Four times 11,025 is 44,100 samples/second, which is the industry standard sample rate used for recording the digital signals of music on compact discs (CDs). We have more to say about sample rates later in this chapter.

OK, to provide illustrations of why it's beneficial to sample an analog audio signal to obtain a digital audio signal, let's look at two simple examples of audio digital signal processing.

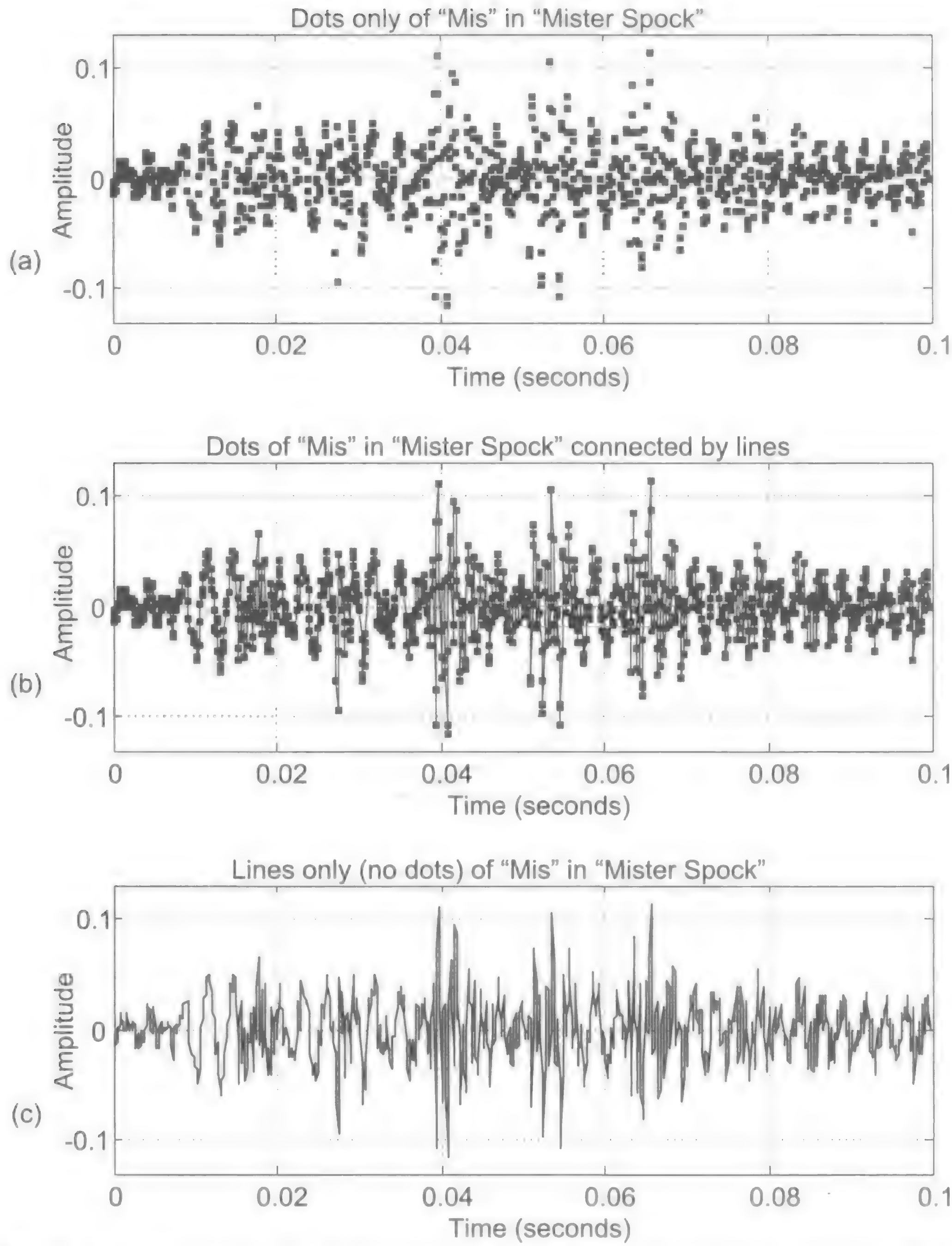


Figure 4-9 Displaying the digital speech signal, "Mis": (a) samples plotted as dots; (b) dots connected by lines; (c) dots connected by lines and dots removed.

AN EXAMPLE OF DIGITAL SIGNAL PROCESSING......

Let's say we're working in a music studio and a pop singer is recording a new song. And say that, unfortunately, during the third verse of the song the singer sings one note *off-key*. Decades ago, the entire song would have to be rerecorded to correct this. And, hopefully, the singer would sing the song perfectly on key during the second recording while maintaining the same emotional intensity as the first recording.

Nowadays, thanks to digital signal processing, correcting performance errors by rerecording entire songs is no longer necessary. Today, the singer's analog voice signal can be sampled by an analog-to-digital converter, and then the digital signal samples are passed to a computer as shown in Figure 4-10. In that figure, the sequence of sampled values (numbers) making up the digital vocal signal are represented by the $x1, x2, x3, \ldots$ notation.

Now let's assume the computer is running commercial software called Auto-Tune®. This software analyzes all the digital samples of the recording by measuring the pitch (frequency) of each musical note sung by the singer. If the software detects

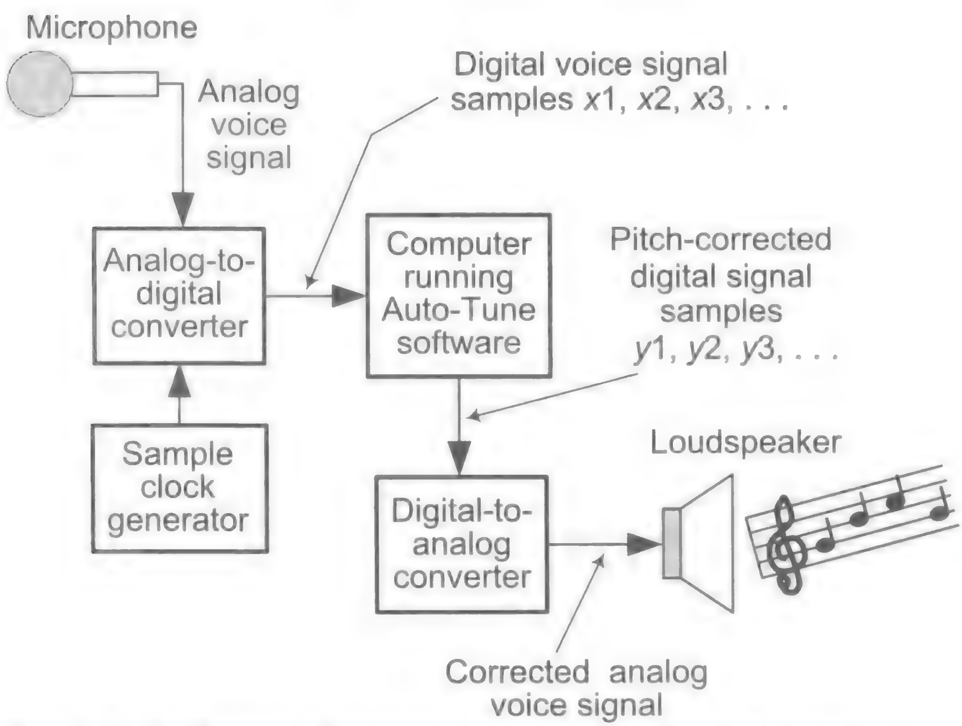


Figure 4-10 Digital audio signal processing by Auto-Tune® software.

a note sung off-key (wrong pitch or frequency), the software determines the on-key note (correct pitch or frequency) that is nearest in pitch to the off-key note. Then, the software replaces the samples of the off-key note with samples of the on-key note.

For example, let's say that the software encounters a sequence of samples of a vocalized note shown by the dots in Figure 4-11(a), and determines that the pitch (frequency) of that note is off-key based on the specified musical key of the song. Next, the software determines the nearest on-key note for that part of the song represented by the dashed-curve waveform in Figure 4-11(a). The software then replaces the off-key samples in Figure 4-11(a) with corrected samples shown in Figure 4-11(b), which represent the nearest on-key note. Notice how the corrected samples in 4-11(b) match the on-key waveform of the dashed curve in 4-11(a).

The *pitch-corrected* $y1$, $y2$, $y3$, . . . digital signal samples of the entire song are then routed to a digital-to-analog converter, which transforms them into an analog signal as shown in Figure 4-10. (We discuss the operation of digital-to-analog converters in a later chapter.) When the analog signal is connected to a loudspeaker, we hear the singer's voice with every musical note perfectly on-key.

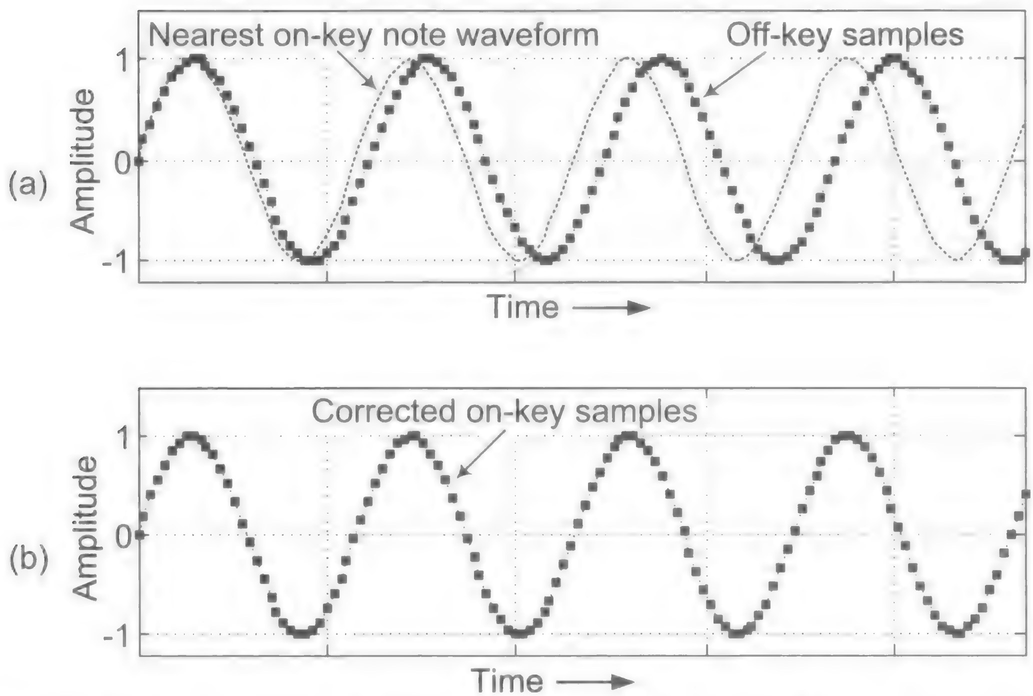


Figure 4-11 Correcting an off-key musical note: (a) samples of an off-key musical note (dots) and the nearest correct note's waveform; (b) samples of the corrected on-key musical note.

This pitch-correction process is used not only in music studios for recording music CDs, but also in live performances in stadiums filled with screaming fans. This type of pitch correction would be impossible without modern digital signal processing.

By the Way

Auto-Tune software makes life easier for people in the music business. That's the good news. The bad news is that you can never again listen to your favorite singer and know whether you're listening to their natural voice or to their Auto-Tune-corrected voice. As it turns out, old vinyl albums with a famous artist occasionally singing off-key are often worth much more money than their *corrected*, re-mastered CD counterparts.

ANOTHER EXAMPLE OF DIGITAL SIGNAL PROCESSING

To understand our second example of why sampling an analog audio signal to obtain a digital signal is beneficial, let's consider how telephones worked in the mid-1900s.

Some of you might be old enough to remember the rotary-dial landline telephone shown in Figure 4-12. To place a call, the user had to lift the handset off the hook, insert a finger in the appropriate numbered hole in the round dial, and rotate the dial clockwise until it encountered a stop. When the user removed his or her finger, a rotary spring spun the dial back to its original position. As the dial spun counter clockwise, an electrical switch was closed and then opened multiple times sending electrical pulses to the telephone company's switching station. (It was similar to quickly flicking a light switch on and off multiple times.) If you started your dialing from the number 4

Figure 4-12 Rotary-dial telephone. (From Bluehand/Shutterstock)

hole in the dial, four electrical pulses were transmitted to the phone company. To make a local phone call you had to repeat this rotary dialing process six more times, once for each of the seven digits of the destination phone's telephone number.

At the telephone company's switching station, electrical hardware would interpret the seven sets of electrical pulses (seven pulses for each number dialed) and automatically, using relay switches, electrically connect your telephone's wires to the destination telephone's wires. For technical reasons, using the early rotary-dial telephones required telephone company operator assistance to make long-distance calls. (Your grandparents well remember the phrase, "Number please.") Be that as it may, the rotary-dial telephone was a great innovation. It eliminated the need for operator assistance for local phone calls.

But technology marched on. Thanks to digital signal processing, making telephone calls became even easier in the early 1960s with the innovation known as the touch-tone telephone that we use today. Modern telephones have a rectangular keypad, shown in Figure 4-13, that we use to make phone calls.

On touch-tone phones, pushing a key activates two internal audio oscillators to generate two distinct analog audio tones whose frequencies depend on which key was pushed. For example, pushing the 8 key generates an 852 Hz audio tone and a 1,336 Hz audio tone that are added together to produce a composite analog audio signal. Likewise, pushing the 4 key generates a 770 Hz audio tone and a 1,209 Hz audio tone that are added together to create a composite analog voltage signal, as shown on the right

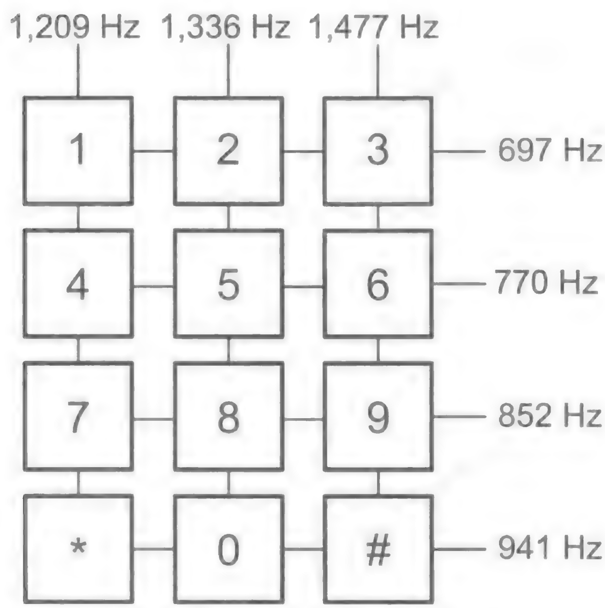


Figure 4-13 Touch-tone telephone keypad.

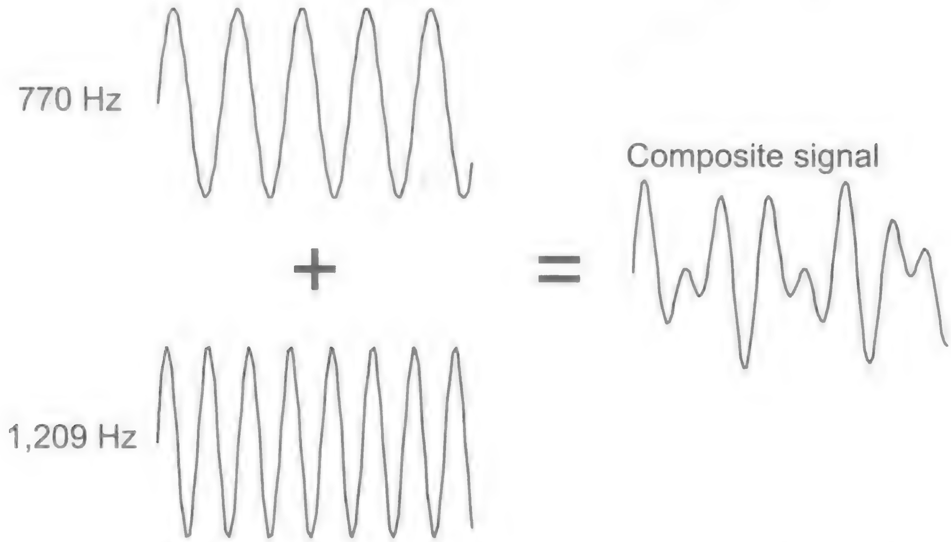


Figure 4-14 Composite audio signal (a 770 Hz tone plus a 1,209 Hz tone) generated when the 4 key is pressed on a touch-tone telephone keypad.

side in Figure 4-14. That composite analog voltage is transmitted over telephone wires to the telephone company's switching station.

When the Figure 4-14 composite analog signal arrives at the telephone company switching station, it is *sampled* by an analog-to-digital converter to generate the digital signal samples represented by the dots in 4-15(b).

That process of analog-to-digital conversion is shown in Figure 4-16. The digital samples are subsequently routed to electronic hardware that performs digital signal processing operations to recognize the frequencies of the two tones contained in the composite signal. This determines which key the telephone caller pressed. The digital signal processing implements an array of tone detectors as shown in Figure 4-16. If the caller presses the telephone's 4 key, the 770 Hz and 1,209 Hz detectors' outputs are activated and the system determines that, indeed, the 4 key was pressed.

So what are the big deals about a touch-tone home telephone and the digital signal processing that takes place in the telephone company's switching station? The big deals are:

- At the switching station, the hardware to detect digital audio tones is *significantly* smaller in size, less expensive, more reliable, faster, and more power efficient than the hardware used to detect old-style, rotary-dial telephone pulses.
- Your touch-tone home phone is smaller in size, lighter in weight, less expensive, and faster in operation than the old-style rotary phones. In addition, no telephone operator assistance is needed for long-distance calls when using touch-tone home phones.

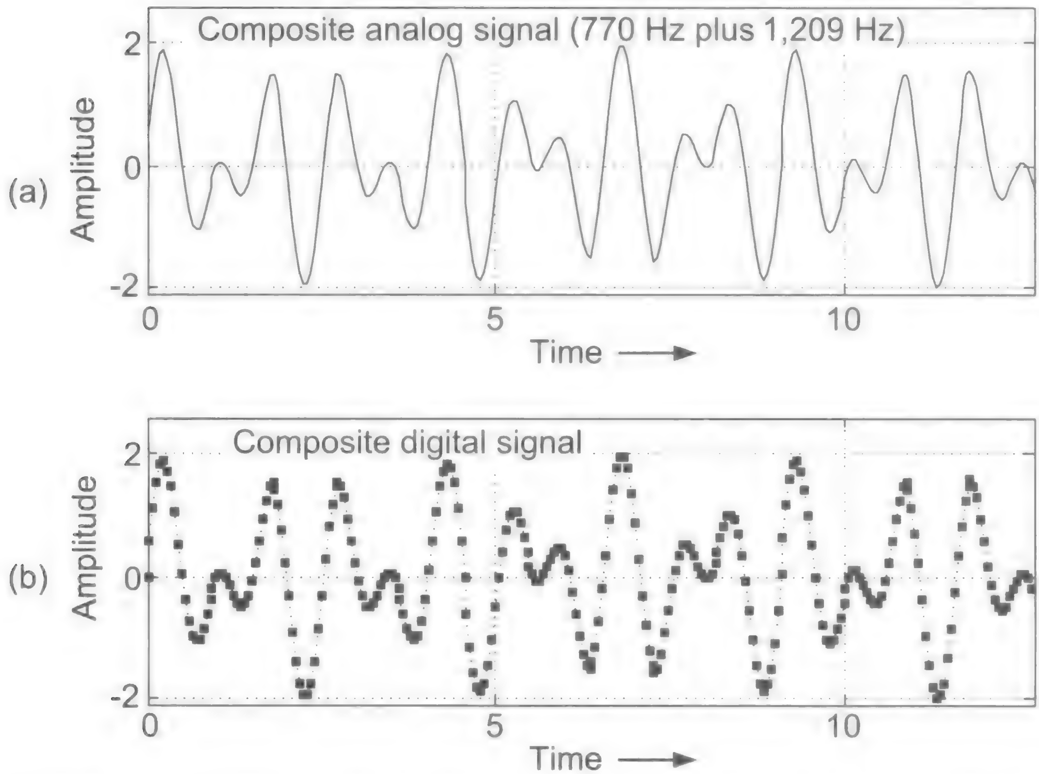


Figure 4-15 Composite audio signal created when the 4 key on a touch-tone telephone's keypad is pressed: (a) analog signal sent to the telephone company; (b) digital signal generated, by sampling, at the telephone company.

By the Way

Digital telephone signals are sometimes relayed by orbiting satellites where the signal is received and then relayed to another continent using high-power on-board amplifiers. These satellites are about 23,000 miles above the equator. Even at the speed of light, it takes about one-quarter of a second (called satellite latency) for the signal's round trip. You notice this on newscasts when the reporter on foreign soil seems to delay his or her answer to a question from the network studio. It's also noticeable in international personal phone calls when you ask, "Do you miss me?" and your sweetheart seems to hesitate before answering!

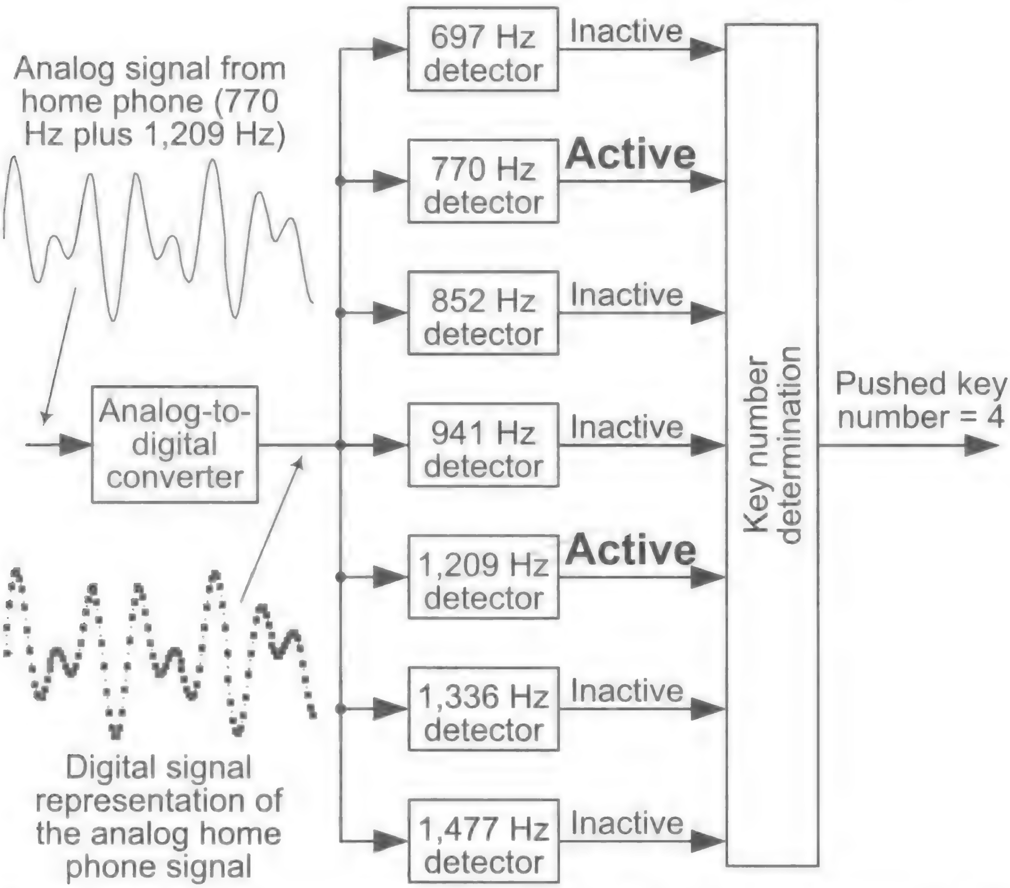


Figure 4-16 Touch-tone telephone pushbutton key recognition process at the telephone company's switching station when the 4 key is pressed.

TWO IMPORTANT ASPECTS OF SAMPLING ANALOG SIGNALS

There are two additional important topics related to sampling an analog signal to generate its corresponding digital signal. We cover these in depth in subsequent chapters, but introduce them briefly here. The first topic is a fundamental restriction imposed on the sample rate used in the analog-to-digital conversion process. The second is the precise characteristics of the numbers produced at the output of an analog-to-digital converter. Let's briefly consider those two topics.

Sample Rate Restriction

As we stated earlier, when we sample an analog signal to generate its corresponding digital signal, as shown in Figure 4-17, we must apply clock voltage pulses to the analog-to-digital converter. The repetition rate of those voltage pulses, measured in samples per second, is called the **sample rate** (or **sample frequency**) of our sampling process. For example, the sample rate of the Figure 4-9 digital speech signal is 11,025 samples/second.

To ensure that the digital signal sequence of numbers, $n1$, $n2$, $n3$, . . . , accurately represents the input analog signal, the sample rate (sampling frequency) of an analog-to-digital converter's clock signal must be greater than twice the frequency of the highest-frequency spectral content of the input analog signal. In the field of digital signal processing, this is called the **Nyquist sampling criterion**. For example, the sample rate of 11,025 samples per second for the digital speech signal in Figure 4-9 is well above twice the highest frequency content of normal conversation.

The origin of this criterion and the ill effects of violating it are so important in the world of digital signal processing that we've dedicated the next chapter, titled "Sampling and the Spectra of Digital Signals," to these topics.

Analog-to-Digital Converter Output Numbers

The second important topic regarding the process of sampling an analog signal is the nature of the numbers produced by an analog-to-digital converter.

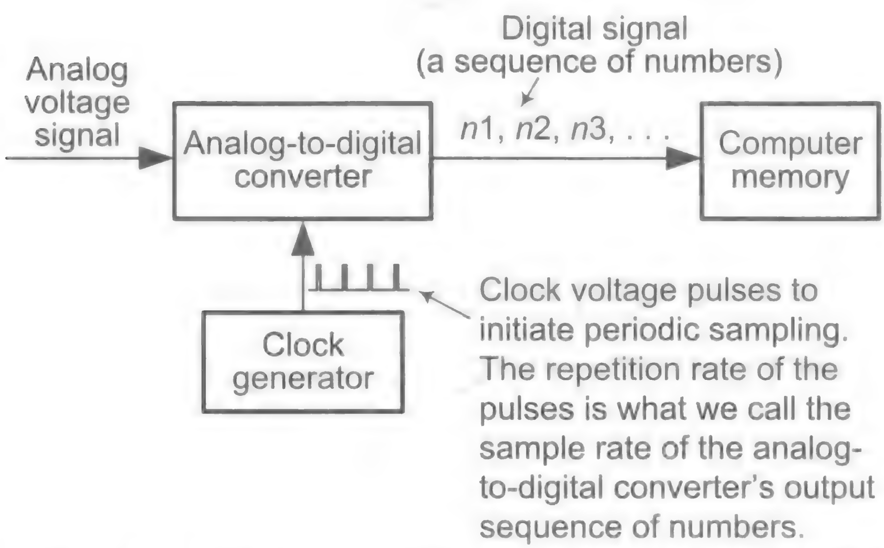


Figure 4-17 Sampling: converting an analog signal to a digital signal.

When we sample an analog signal, as shown in Figure 4-17, the digital signal's sequence of numbers is not in the form of decimal numbers that we're so familiar with in our daily lives. The digital signal's sequence of numbers, $n1$, $n2$, $n3$, . . . , are in the form of what we call **binary numbers**. The interesting topics of binary numbers and why we use them are discussed in Chapter 9.

SAMPLE RATE CONVERSION ....

When reading the literature of digital signal processing or listening to signal processing engineers, you may encounter the terms **decimation** and **interpolation**. Those terms refer to changing the sample rate of a digital signal.

That might seem like a strange idea but changing the sample rate of a digital signal, known as **sample rate conversion**, has many applications. In fact, both decimation and interpolation take place whenever you use your cell phone or smartphone. In the following sections, we briefly describe the processes of both decimation and interpolation.

Decimation

Regardless of its dictionary definition, for us the term **decimation** means to reduce the sample rate of a digital signal. We'll explain this idea with an example.

Consider the 250 Hz sine wave voltage signal shown in Figure 4-18(a). If we sampled that analog signal using an analog-to-digital converter, at a sample rate of 3,000 Hz the resulting digital signal's samples would be those shown in Figure 4-18(b). If, for some reason, we wanted a digital version of the original analog signal having a sample rate of 1,000 Hz, we would not need to repeat an analog-to-digital conversion process. We could simply decimate the Figure 4-18(b) digital signal by a factor of 3. That decimation-by-3 process means we simply retain every third sample of the Figure 4-18(b) signal, and discard the remaining samples. Retaining only every third sample of the Figure 4-18(b) signal results in our desired 1,000 sample rate digital signal shown in Figure 4-18(c).

Interpolation

Interpolation refers to the process of increasing the sample rate of a digital signal. As we did with decimation, let's look at that idea by way of an example.

Figure 4-19(a) shows a digital signal whose sample rate is 1,000 Hz. Suppose we want that signal to have a sample rate of 3,000 Hz (higher-frequency sampling rate). The first step in our interpolation-by-3 process is to create a modified digital signal by inserting two zero-valued samples in between each of the original digital signal's

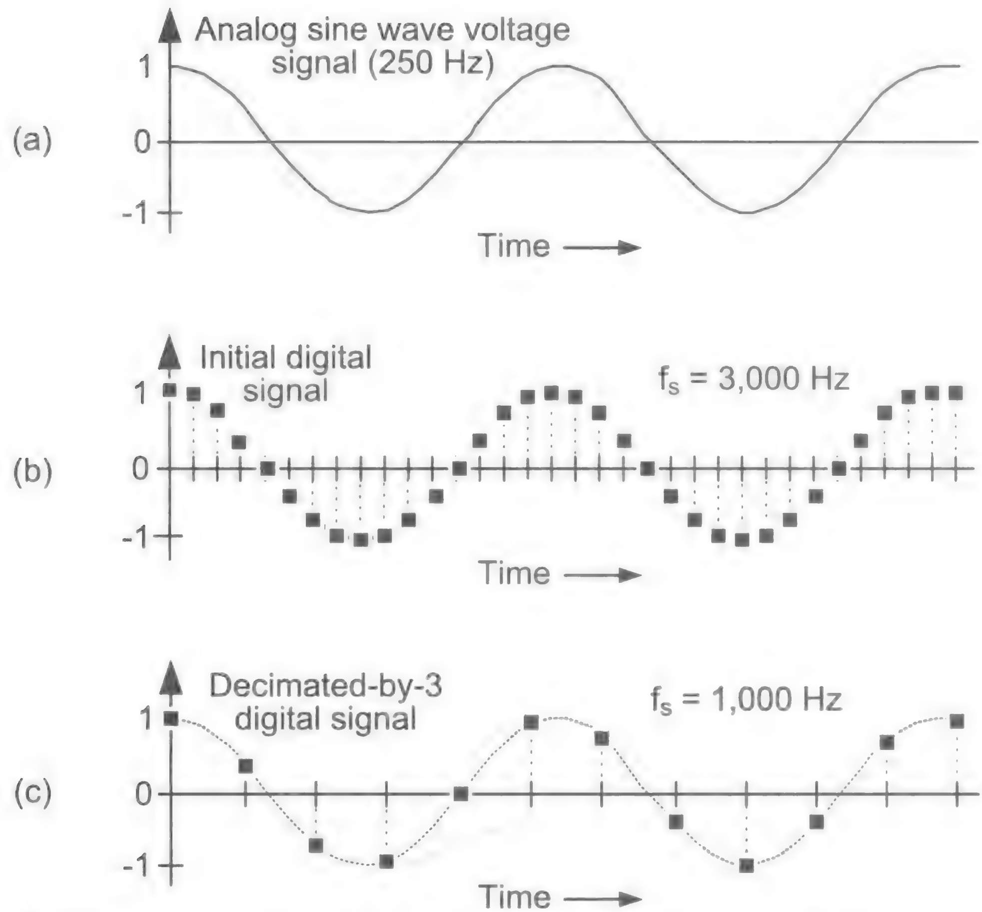


Figure 4-18 Decimation by a factor of 3: (a) original 250 Hz analog sine wave voltage signal; (b) initial digital signal having a sample rate of 3000 Hz; (c) decimated-by-3 digital signal having a sample rate of 1,000 Hz.

samples. That new digital signal is shown in Figure 4-19(b), where the zero-valued samples are represented by the circular dots. The sample rate of the Figure 4-19(b) signal is now 3,000 Hz as indicated in Figure 4-19(c). Our final step is to pass the Figure 4-19(b) signal through a digital **lowpass filter** whose output signal is our desired interpolated-by-3 signal shown in Figure 4-19(d). A **lowpass filter** is a process that allows low-frequency signal energy to pass, but blocks high-frequency signal energy. We discuss the behavior and implementation of digital lowpass filters in Chapter 8.

Where decimation by 3 was a single-step process of simply discarding time samples, interpolation by 3 is a two-step process of zero-valued sample insertion followed by lowpass filtering.

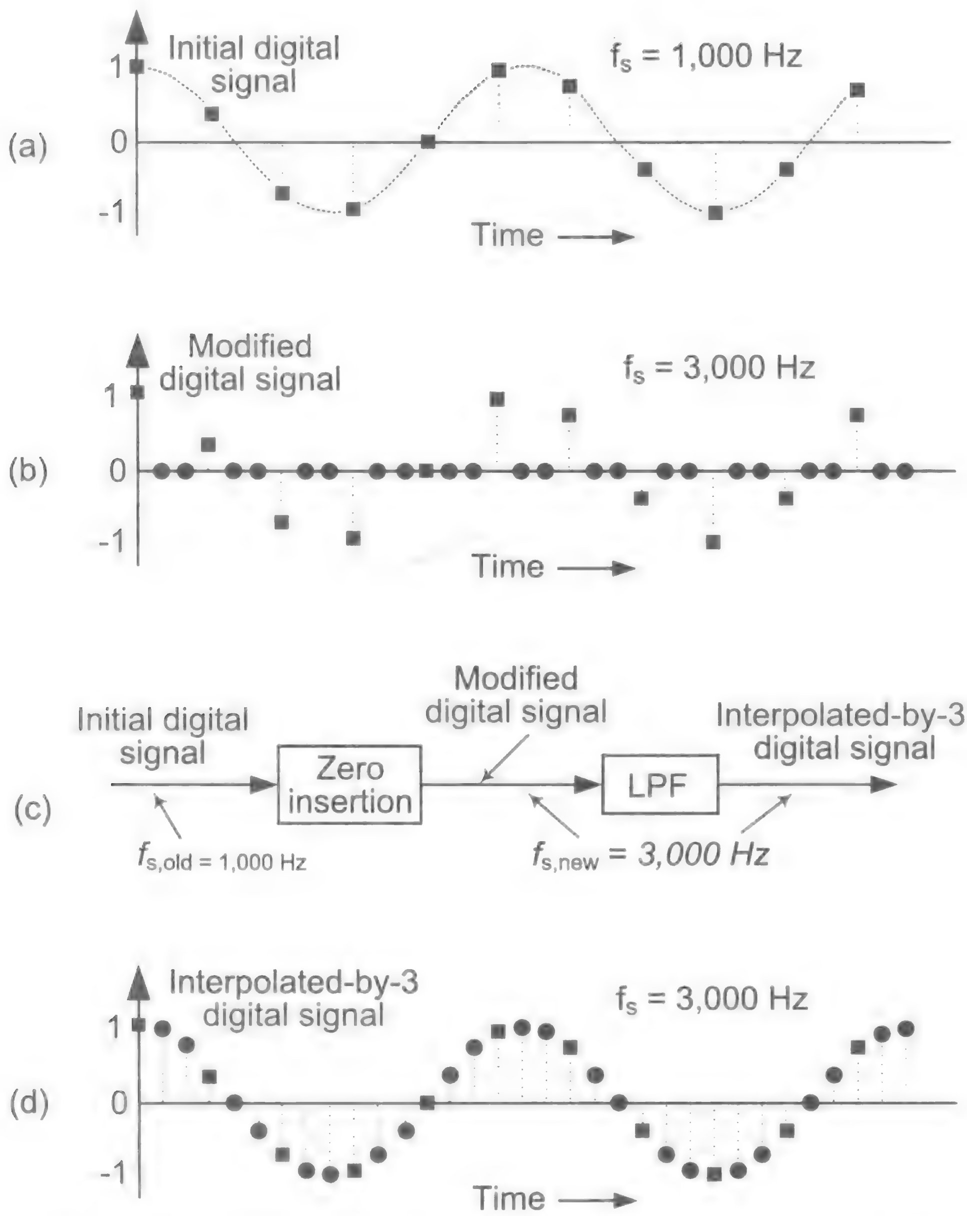


Figure 4-19 Interpolation by a factor of 3: (a) original digital signal having a sample rate of 1,000 Hz; (b) modified digital signal with zero-valued samples inserted; (c) interpolation process signal flow; (d) interpolated-by-3 digital signal having a sample rate of 3,000 Hz.

WHAT YOU SHOULD REMEMBER

The concepts you should remember from this chapter are:

- The phrase *digital signal* has two different meanings:
  - Definition #1: An analog voltage signal that alternates between two distinct voltage values (see Figure 4-1).
  - Definition #2: A sequence of discrete, individual, numbers (see Figure 4-5.)

  We use definition #2 in this book.
- Digital signals, discrete sequences of numbers, can be stored in the memory of a computer.
- Digital signals are produced in three ways:
  - Observing and collecting meaningful data (see Figure 4-6(a))
  - Computer software
  - Sampling an analog signal using analog-to-digital converter hardware (see Figure 4-7 and Figure 4-8)
- Digital signals have a sample rate associated with them. The sample rate of a digital signal is the repetition rate of the signal's samples measured in samples per unit of time. For example, the sample rate for the digital signal in Figure 4-5 is one sample per month. The sample rate for the digital signal in Figure 4-6(a) is one sample per day. The sample rate for the digital signal in Figure 4-9(a) is 11,025 samples per second.
- To ensure that a digital signal sequence accurately represents an analog signal, the sample rate (sampling frequency) of an analog-to-digital converter's clock signal must be greater than twice the frequency of the highest-frequency spectral content of the analog signal.
- The term *decimation* refers to the process of decreasing the sample rate of a digital signal. The term *interpolation* refers to the process of increasing the sample rate of a digital signal.

5 Sampling and the Spectra of Digital Signals

This chapter explains what digital signal processing engineers mean when they talk about the topics of sampling and the *spectra of digital signals*. Both subjects are important as well as interesting and, after reading this chapter, you'll understand the meaning of this terminology.

In Chapter 2, we introduced the idea of analog signals. These are signals whose voltage amplitudes smoothly change in value as time passes. Then in Chapter 3, we discussed the important notion of the spectrum of an analog signal, which is a description, or measure, of the frequency content of an analog signal. Although you're usually not concerned with the spectra of analog signals in your daily life (unless you're tuning a guitar or a piano), spectrum analysis is of profound and fundamental importance to signal processing engineers. In Chapter 4, we described the idea of a digital signal, a sequence of numbers faithfully representing an analog signal. In this chapter, we examine the concept of the spectra of digital signals. Familiarity with these spectra is essential for anyone wanting to understand the fundamentals of digital signal processing.

And as it turns out, with respect to spectral content, surprising things happen when we convert an analog signal to its digital signal representation. This chapter explains the important characteristics of such digital spectra. With that said, let's briefly review the key aspects of analog signal spectra in preparation for a solid understanding of digital signal spectra.

ANALOG SIGNAL SPECTRA—A QUICK REVIEW ....

In Chapter 2, we looked at analog signals whose voltage waveforms vary in amplitude as time passes. For example, let's say an analog engineer's job is to design an electronic circuit that generates a 1,000 Hz sine wave audio tone for use as a beeper

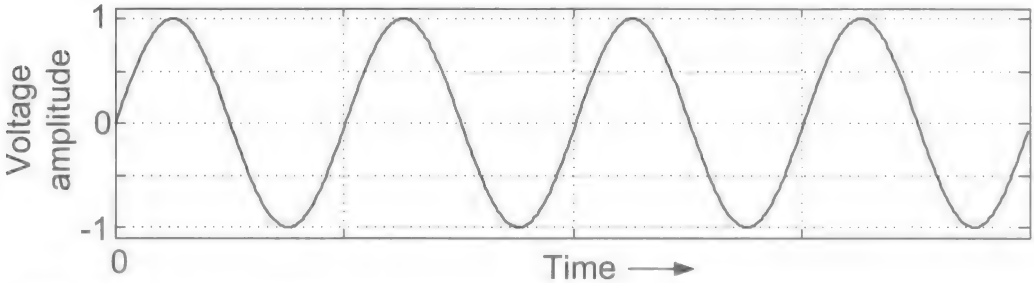


Figure 5-1 Time waveform of a sine wave voltage.

for a microwave oven. The engineer solders the appropriate electronic components to a printed circuit board in order to generate a sine wave, shown in Figure 5-1, whose voltage value smoothly fluctuates 1,000 times per second. If the engineer used speaker wire to connect that sine wave voltage to a loudspeaker, he'd hear a pure audio tone.

At that point, the audio tone generation task is only half finished. The engineer must next ensure that the spectrum, the frequency content, of his signal is indeed 1,000 Hz. (If it were 25,000 Hz, only his dog would hear it.) So the engineer applies the sine wave voltage to the input of a spectrum analyzer to display the spectrum of the audio signal. If the spectral display is that shown in Figure 5-2, the sine wave has the correct frequency spectrum and the engineer's job is finished. The point of this little story is that not only is the time waveform shape of the audio signal important, but the spectrum of the signal is equally critical.

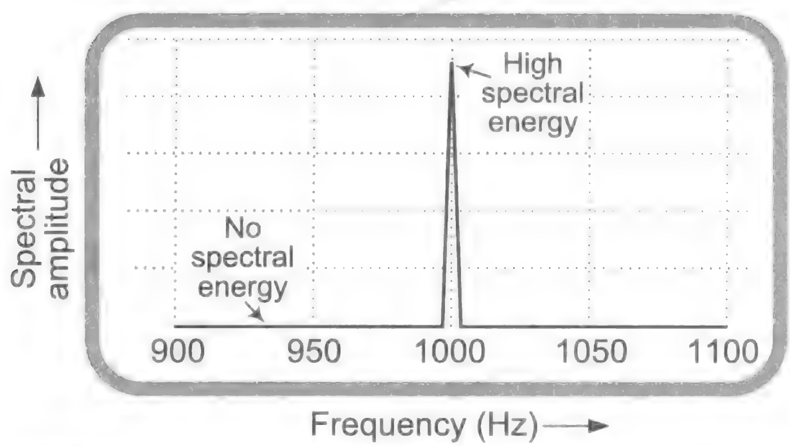


Figure 5-2 Spectrum of a 1,000 Hz sine wave voltage.

Another example of the importance of signal spectra involves commercial AM (amplitude modulation) radio broadcast stations. In the United States, AM radio broadcast systems are carefully designed so that their transmitted signals never have a spectral bandwidth greater than 10 kHz (10,000 Hz). In addition, the radio stations' **center frequencies** (also called **carrier frequencies**) are carefully controlled so that they are separated by at least 10 kHz.

Figure 5-3 represents a hypothetical portion of the AM broadcast band. Radio Station #4's center frequency is 640 kHz, separated by 10 kHz from the center frequencies of its neighboring stations as shown in Figure 5-3. To listen to Radio Station #4, we would tune our AM radio to a frequency of 640 kHz (or as the radio announcer would say, "Welcome to 640 on your radio dial").

In Chapter 2, we stated that the bandwidth of AM radio audio signals is restricted to 5 kHz. That's true; however, the inherent behavior of the amplitude modulation (AM) process, necessary so that signals can be transmitted using transmission antennas, doubles that audio signal's bandwidth such that the radiated (broadcasted) signal has a bandwidth of 10 kHz as we see in Figure 5-3.

That 5 kHz audio bandwidth restriction is critical because if an equipment malfunction occurred at Radio Station #4 and the audio bandwidth was inadvertently 8 kHz, the radiated signal's bandwidth would be 16 kHz as shown in Figure 5-4. That situation would cause Radio Station #4's radiated signal to interfere with neighboring radio stations. The result would be that if you tuned your AM radio to Radio Station #3 or Radio Station #5, you'd hear some of the high-frequency audio from Radio Station #4 in the background of your desired station's audio signal. Radio station engineers ensure this scenario never occurs.

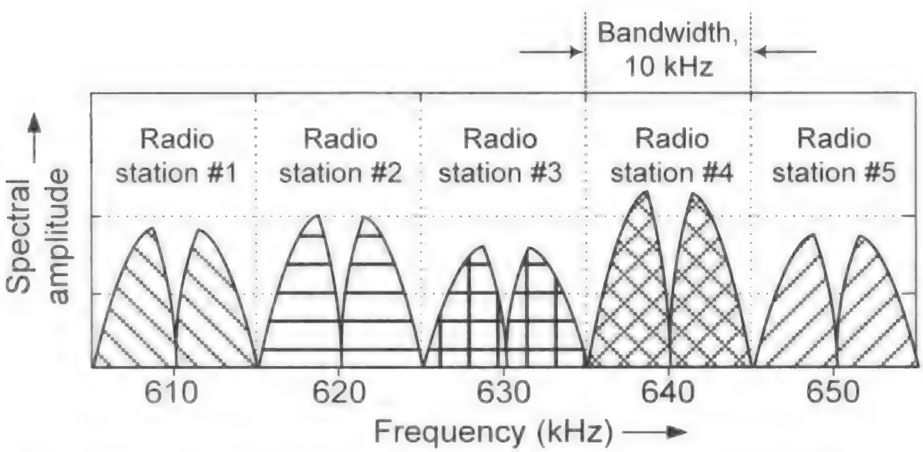


Figure 5-3 Spectrum of a portion of the AM radio broadcast band.

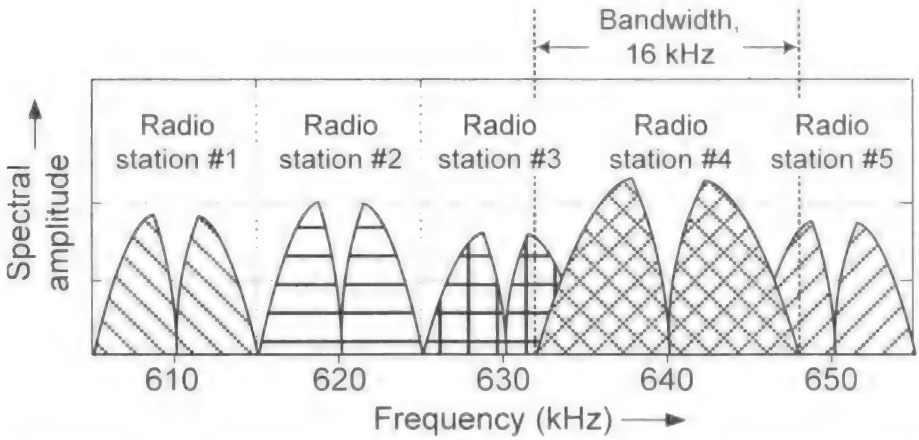


Figure 5-4 AM radio interference when Radio Station #4's transmitted signal bandwidth exceeds 10 kHz.

By the Way

The allocation of radio transmission center frequency and bandwidth has been a hot topic for many years. In the United States, the Federal Communication Commission (FCC) uses hearings, auctions, and even lotteries to allocate this finite and valuable resource. This is a far cry from the early days of communications in the 1800s when Morse code signals were sent using a spark-gap transmitter that radiated over very wide and unpredictable frequencies.

The restriction that an AM radio station's transmitted signal center frequencies must be separated by exactly 10 kHz is also critical. Let's say a transmitter equipment malfunction occurred at Radio Station #4 and its radiated signal's center frequency was inadvertently 643 kHz as shown in Figure 5-5. That situation would cause Radio Station #4's radio signal to interfere with Radio Station #5. To prevent one radio station's signal from encroaching on another station's signal, AM radio station engineers are tasked to see that the scenarios in Figure 5-4 and Figure 5-5 *never* occur. (Appendix C provides additional information concerning AM radio signals.)

The purpose of the discussion above is to show why we care so much about signal spectra. We cannot overemphasize the importance of controlling and measuring signal spectra in the practical world of signal processing. Having said that, let's now look at the spectra of digital signals.

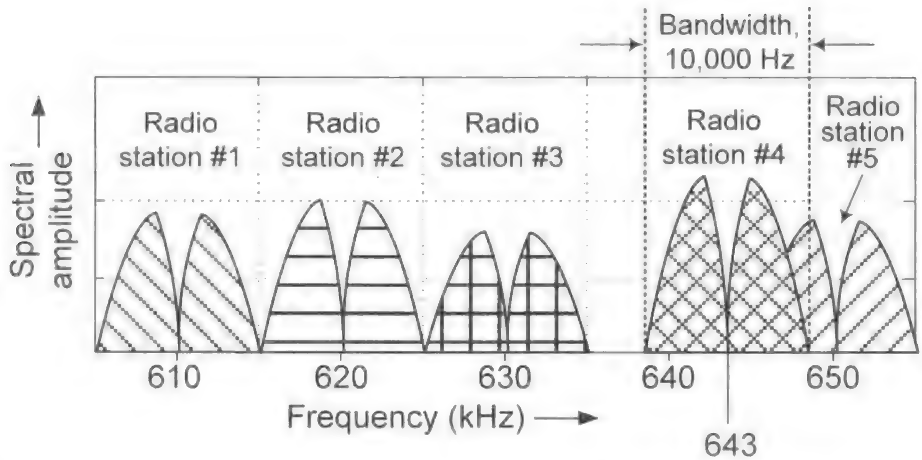


Figure 5-5 AM radio interference when Radio Station #4 transmits at the wrong center frequency.

HOW SAMPLING AFFECTS THE SPECTRA OF DIGITAL SIGNALS

When we convert an analog signal to a digital signal, the spectrum of the digital signal depends on two things: (1) the spectrum of the analog signal and (2) the f_s sample rate of the analog-to-digital conversion process. This section explains that two-fold dependence.

The vast majority of digital signals are generated by the periodic sampling process that we introduced in the last chapter. That process is shown in in Figure 5-6. We've seen the diagram in that figure before. An analog signal is applied to an analog-to-digital converter, whose sample rate is f_s samples per second (commonly referred to as f_s Hz), producing our digital signal, a sequence of discrete numbers.

Our goal in this sampling process is to generate a digital signal that includes *all* of the information contained in the input analog signal. For example, if the analog input signal is a music signal, we may want to assemble the digital signal's samples (numbers) into a data file and attach that file to an e-mail and send it to a friend. Using the appropriate software on her home computer, our friend can use the digital signal data file to reproduce the original analog music signal and listen to it on her computer's loudspeakers.

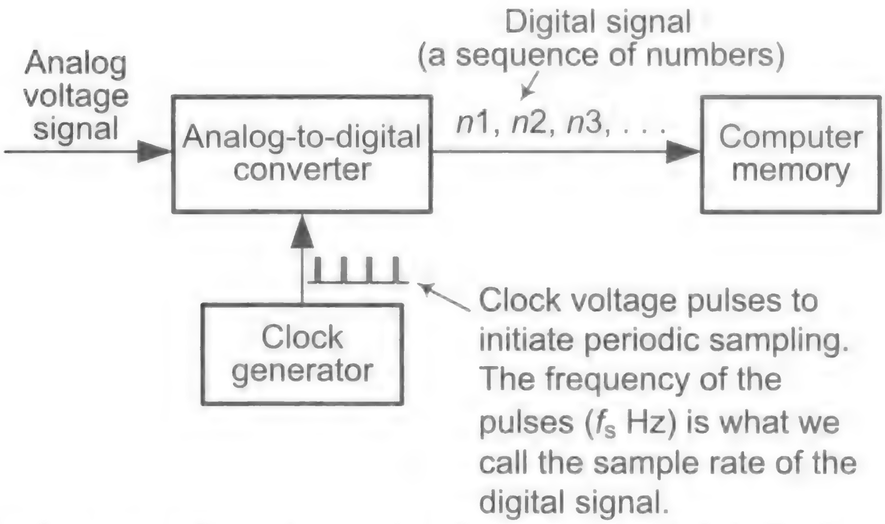


Figure 5-6 Generating a digital signal (the sequence of numbers *n*1, *n*2, *n*3, . . .) by sampling an analog signal.

Thinking again about the notion of spectra, the analog voltage signal at the input of our sampling process will have some fixed spectrum. For example, if the analog signal is the output of a microphone, the signal may have a spectrum like that depicted in Figure 5-7.

The surprising thing about the periodic sampling process in Figure 5-6 is that the spectrum of the digital signal's sequence of numbers not only depends on the spectrum of the analog input signal, but the spectrum of the digital signal also depends on the frequency of the f_s sample rate! At first glance, that dependence may not seem too important, but indeed it is. If the analog input signal in Figure 5-6 has the spectrum

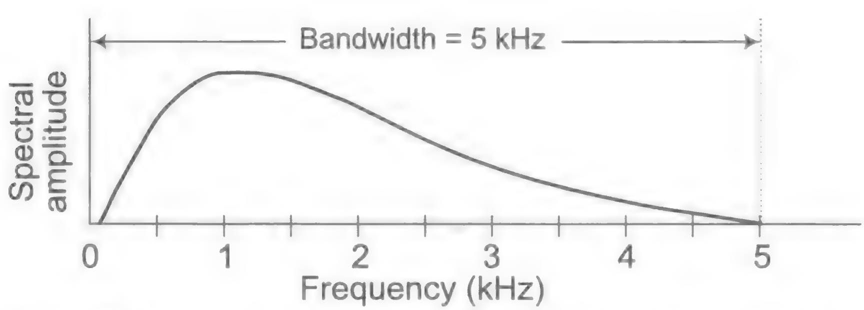


Figure 5-7 A typical spectrum of the analog output voltage of a microphone.

shown in Figure 5-7, then we want the digital signal to have that same spectrum. In that case, the digital signal contains all the information, undistorted, that existed in the analog signal. However, it's possible that, with an incorrectly chosen f_s frequency for our clock pulses in Figure 5-6, the digital signal's spectrum will not be the same as the analog signal's spectrum. When that happens, the digital signal is a corrupted version of the input analog signal.

What we're saying here is that with an improper f_s sample rate, an analog music signal converted to a digital signal and e-mailed to a friend would sound like unintelligible audio garble on the friend's computer loudspeakers. So, following that warning with regard to poorly chosen f_s sample rates, let's now take a closer look at what situations can cause unexpected, and possibly detrimental, digital signal spectral problems.

The Mischief in Sampling Oscillating Quantities

Before we examine the spectral effects of sampling analog signals, we'll engage in another thought experiment.

Suppose we have a mechanical clock that has a second hand but no minute or hour hand. The second hand makes a full 360 degree rotation every 60 seconds. Next, suppose we take a photograph of the clock when the second hand is pointed at 12:00 noon and then take additional photos every 55 seconds. Our first four photographs would look like those in Figure 5-8(a) through Figure 5-8(d). We can think of those photographs as *samples* of the smooth, continuous rotary motion of the clock's second hand.

Upon showing those time-ordered photos in Figure 5-8 to someone, what would he or she think about the direction of motion of the second hand as time advances? That's right; from the photos, it appears that the second hand is rotating in the counterclockwise direction! Now, if we took our photographs much more often, say once every 5 seconds, the time-ordered photos would look like those in Figure 5-9(a) through Figure 5-9(d). And in that figure, the sequence of photos correctly shows that the second hand is rotating in the clockwise direction.

What we learn from our thought experiment is that sampling the clock's continuously rotating second hand too slowly (one photo every 55 seconds) yields misleading results. Likewise, sampling the rotating second hand sufficiently often (one photo every 5 seconds) produces correct results. Further experimentation with photographs of our clock's second hand shows us that the words *sufficiently often* mean that if we take photos *more* often than every 30 seconds, the sequence of photos would show the correct clockwise motion of the second hand. (If we took the photos exactly at 30-second intervals, the hand would alternate between the top and bottom of the clock, and we could not tell its direction of rotation.)

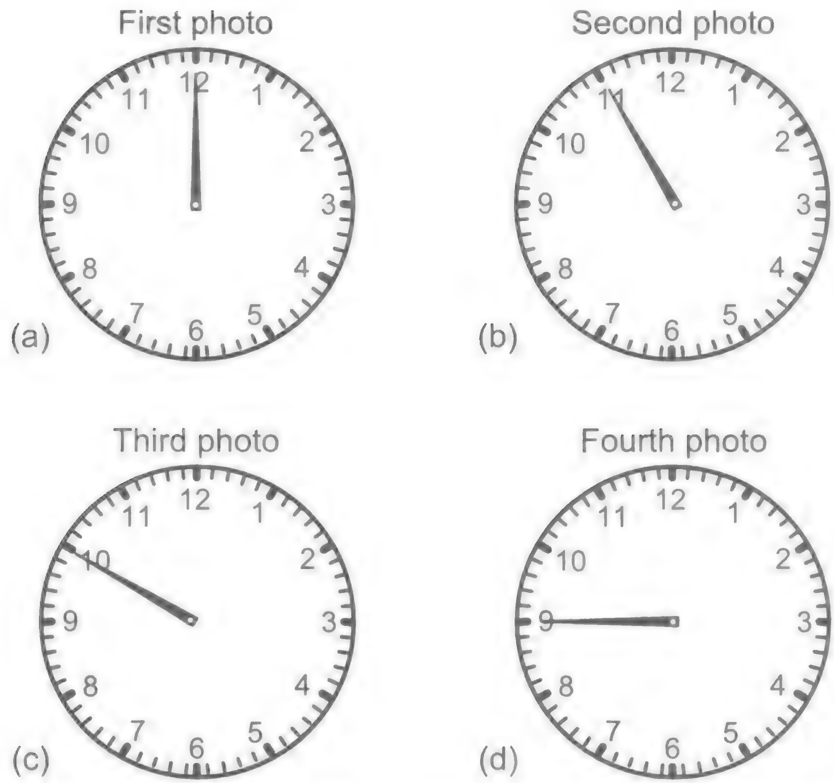


Figure 5-8 *Periodic photos, one photo every 55 seconds, of a clock's rotating second hand.*

So, from this simple thought experiment we can state one of the most important principles in all of digital signal processing. That is:

> To correctly represent a continuous (analog) periodic phenomenon whose cyclic period duration is t seconds with a sequence of samples, the time between samples must be less than half of t. For our clock, t is 60 seconds, so we must sample at intervals of less than 30 seconds.

Stated differently:

> To correctly represent a continuous (analog) periodic phenomenon whose frequency is f cycles per second with a sequence of samples, the f_s sample rate must be greater than two times f.

That last statement may not seem terribly profound, but it most certainly is. This sample rate restriction pervades essentially every aspect of digital signal processing.

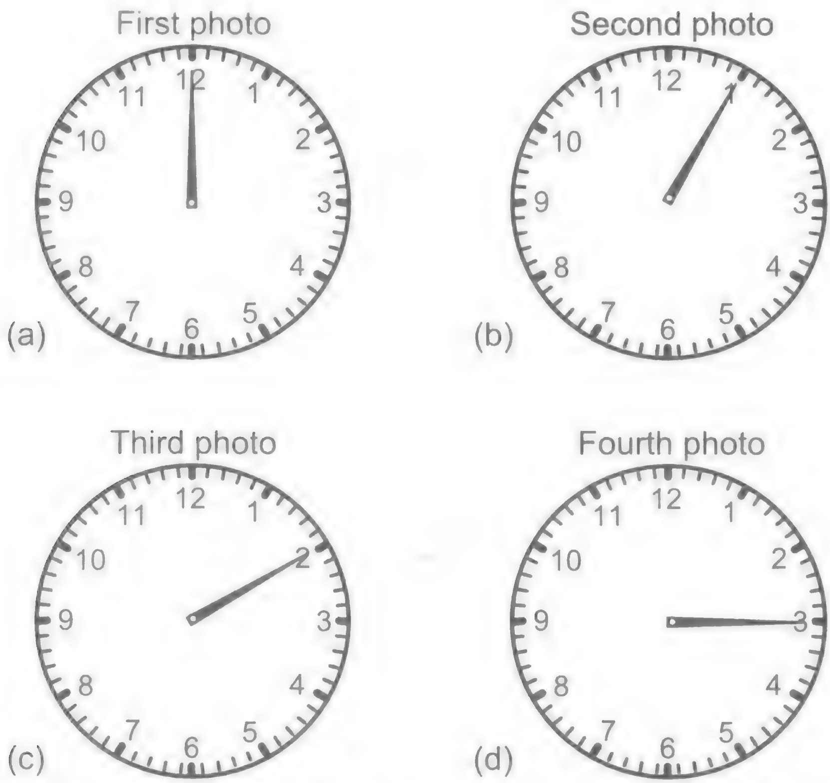


Figure 5-9 Periodic photos, one photo every 5 seconds, of a clock's rotating second hand.

By the Way

You've seen that counterclockwise rotation illusion in Figure 5-8 before. In the old Western movies, recall that a wagon's wheels sometimes appeared to rotate backward. That illusion occurred because the movie camera was taking a sequence of snapshots—a new snapshot 24 times per second. The spokes of a rotating wheel behave exactly like the clock's second hand in our Figure 5-8 presentation. Depending on the rotation rate of the wagon wheel (in revolutions per second), its spokes may have appeared to be spinning far too slowly compared to the forward speed of the wagon. And in some case, the spokes even appeared to be rotating backward, the effect we saw in Figure 5-8.

Your cell phone's video camera takes 20 to 30 pictures per second. You might try recording a cell phone video of a spinning electric fan, and turn the fan off while recording your video. Upon viewing your finished video, for a few moments the fan blades will appear to spin backward.

Next, we relate the sample rate restriction discussed above to the process of sampling an analog sine wave voltage in our quest to understand the spectra of digital signals.

Sampling Analog Sine Wave Voltages

In the last section, we introduced the notion that to correctly represent the behavior of a smoothly changing quantity (the position of a clock's second hand) with samples, we must capture those samples (take photos of the second hand) at a sufficiently high sample rate. Let's now relate that idea to sampling a simple analog sine wave voltage signal. Once we understand the behavior of sampling a sine wave signal, we'll be able to understand more complicated examples of sampling, like the sampling of an audio signal on a music **compact disc (CD)**.

Correct Sampling of an Analog Sine Wave

The process of sampling an analog sine wave voltage signal is depicted in Figure 5-10. An analog sine wave voltage is connected, by way of a two-wire cable, to the input of the analog-to-digital converter hardware. The output of the converter is a digital signal (a sequence of numbers) that is routed, by way of a multiwire cable, to a computer's memory. Because the sample rate of this process is 1,000 Hz, 1,000 numbers per second are transferred to the computer. Using computer software, we can examine and display the values of our digital signal's samples.

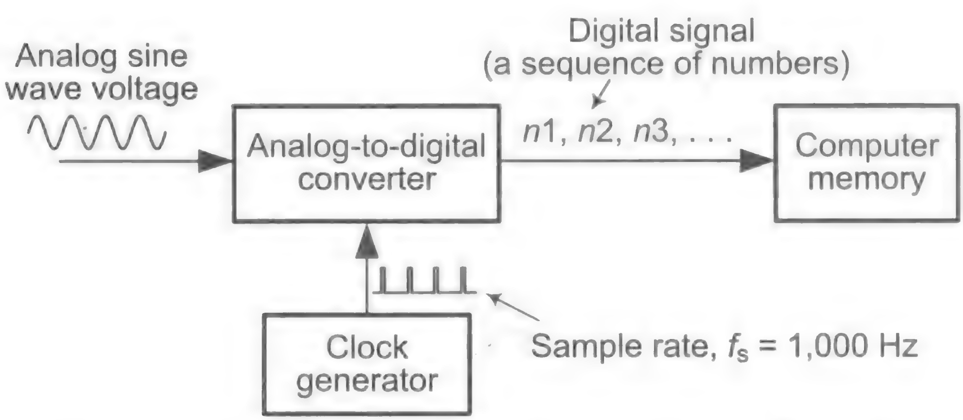


Figure 5-10 Sampling an analog sine wave voltage signal. The f_s sample rate is 1,000 Hz.

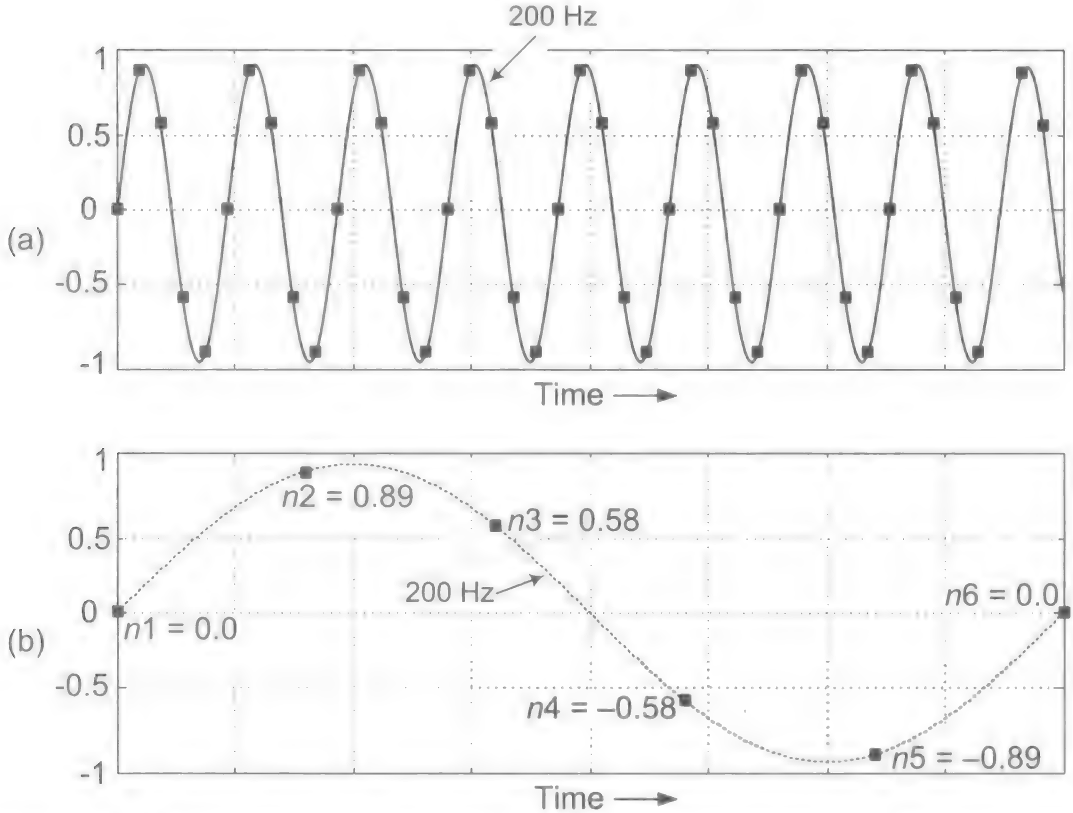


Figure 5-11 *Sampling an analog sine wave voltage signal when the f_s sample rate is 1,000 Hz: (a) sampling a 200 Hz sine wave; (b) the first six sample values.*

When the input analog sine wave voltage's frequency is 200 Hz, the analog sine wave is shown by the solid curve in Figure 5-11(a) and the dots represent the discrete numbers produced by the analog-to-digital converter and stored in the computer's memory.

If we zoom in on the first six samples of our digital signal, we see those sample values in Figure 5-11(b). For reference purposes, we include the dotted sine wave curve in Figure 5-11(b). There's really nothing new here. This is merely the process of sampling an analog sine wave voltage, a process we first learned about in the last chapter.

We have *correctly sampled* Figure 5-11's analog sine wave because our 1,000 Hz sample rate was greater than two times the frequency of the 200 Hz sine wave. We generated more than two discrete samples for each complete cycle of the analog sine wave. To be thorough, we list the first six samples of our digital signal in Table 5.1.

Table 5.1 First Six Values of the 200 Hz Digital Signal

Digital Signal Sample Designator	Sample Value
*n*1	0
*n*2	0.95
*n*3	0.59
*n*4	–0.59
*n*5	–0.95
*n*6	0

Incorrect Sampling of an Analog Sine Wave

Let's experiment a little: If we change the frequency of our input analog sine wave voltage from 200 Hz to 800 Hz, the first six samples of our digital signal would be the dots shown in Figure 5-12(a). And, if we change the frequency of the input analog sine wave from 800 Hz to 1,200 Hz, the first six samples of our digital signal would be the dots shown in Figure 5-12(b). Next, look very carefully at Figures 5-11(b), 5-12(a), and 5-12(b). Do you see anything interesting?

That's right; the three digital signals (the three sets of dots) in those figures are identical! We verify this astounding situation by plotting the 200 Hz, 800 Hz, and 1,200 Hz sine waves and their digital representations in the rather busy Figure 5-13. *From the discrete sample values alone, we cannot determine the original sine wave's frequency.* This situation is arguably the most astonishing consequence of dealing with sampled data.

The situation where a digital signal obtained from sampling a high-frequency analog sine wave takes on the identity of a digital low-frequency sine wave is called **aliasing**. Just as criminals may take on a new identity and a new name (an alias) to appear to be someone they are not, the 800 and 1,200 Hz digital sine wave signals appear identical to the sampled 200 Hz digital signal. This happens because we have not correctly sampled the 800 and 1,200 Hz analog sine waves. We *did not* generate more than two discrete samples for each complete cycle of those higher-frequency analog sine waves.

Figure 5-14 shows the incorrect sampling of the 800 and 1,200 analog sine waves. In the figure, the bold curves show complete cycles of the two analog sine waves where only one discrete sample is generated during those individual cycles.

Again, the point here is: from our digital signal alone (the sample values listed in Table 5.1), we cannot tell if that digital signal was obtained from sampling a 200, 800, or 1,200 Hz analog sine wave.

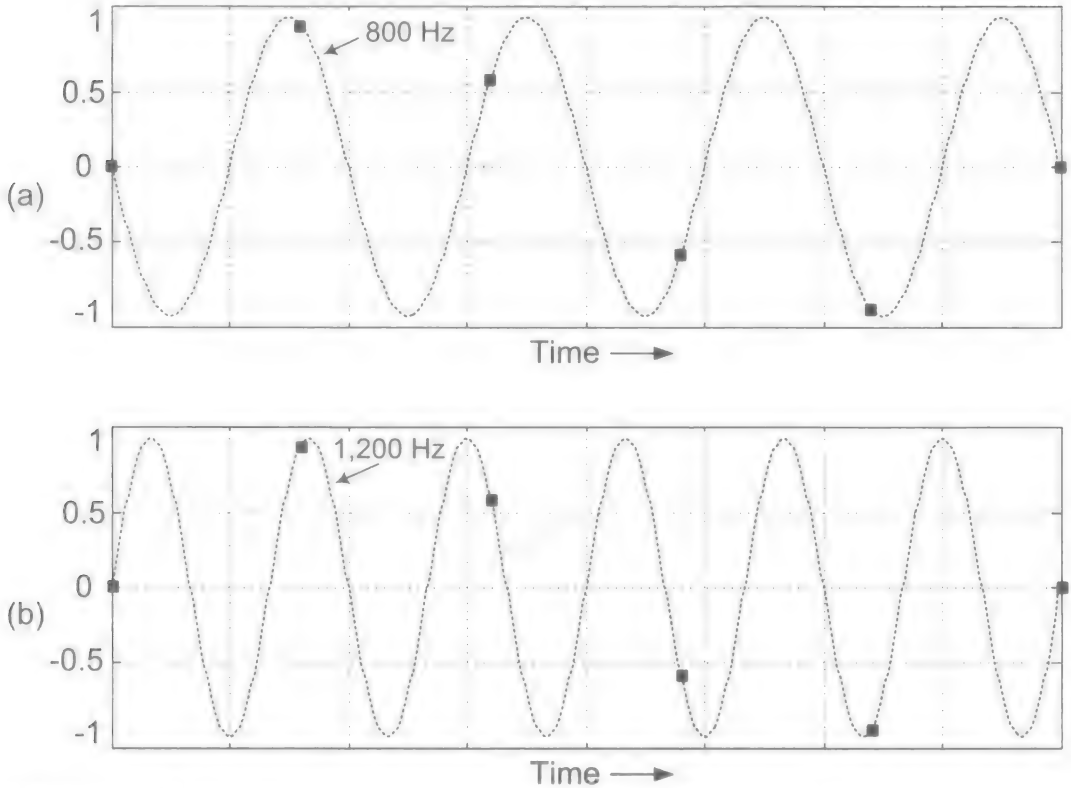


Figure 5-12 Sampling sine wave signals of different frequencies: (a) sampling an 800 Hz sine wave; (b) sampling a 1,200 Hz sine wave.

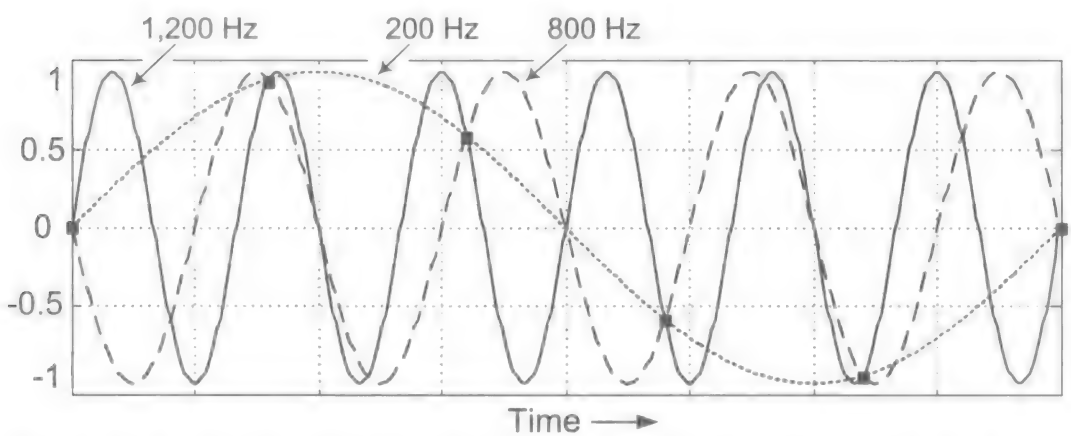


Figure 5-13 Sampling 200 Hz, 800 Hz, and 1,200 Hz sine waves when the f_s sample rate is 1,000 Hz.

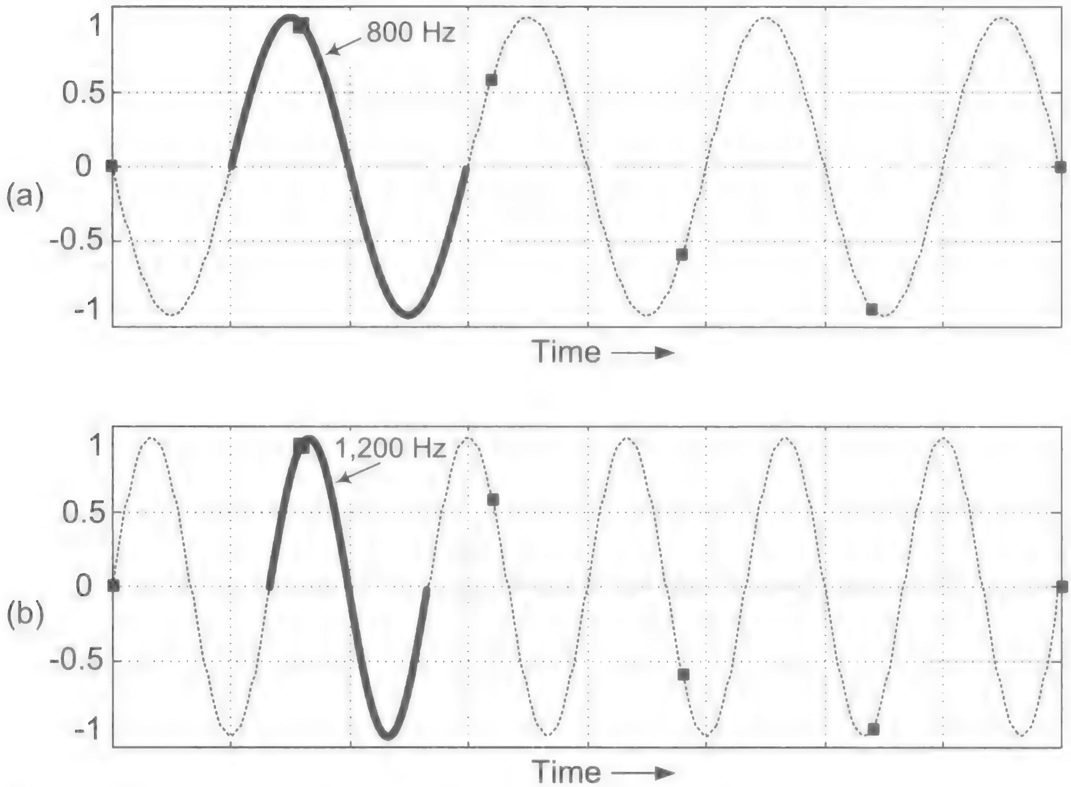


Figure 5-14 Incorrect sampling showing where a complete analog cycle is represented by less than two samples: (a) an 800 Hz sine wave; (b) a 1,200 Hz sine wave.

Why We Care about Aliasing

The situation where a digital signal can be the sampled representation of multiple sine waves having different frequencies, what we call **aliasing**, is of critical importance in the field of digital signal processing. We can easily demonstrate this importance with an example.

Earlier in this chapter, we mentioned the notion of converting an analog music signal into a digital signal, assembling the digital signal's numerical samples into a data file, and e-mailing that file to a friend. Using appropriate software and the built-in digital-to-analog converter on her home computer, the friend can convert the digital signal file to an analog music signal and listen to it on her computer's loudspeaker.

However, doing this with the digital signal in Figure 5-13 can lead to problems as we show in Figure 5-15. If the e-mailed digital signal was a sampled 200 Hz audio tone (sampled at a rate of 1,000 Hz), our friend would hear a 200 Hz tone from her computer's speakers as shown in Figure 5-15(a). Things are as they should be.

On the other hand, if the original sampled analog signal were an 800 Hz audio sine wave as shown in Figure 5-15(b), our friend would still hear the reproduced

analog sine wave as being a 200 Hz audio tone! Likewise, if the original sampled analog signal were a 1,200 Hz audio sine wave as shown in Figure 5-15(c), our friend would again hear the reproduced analog sine wave as being a 200 Hz audio tone. The scenarios in Figure 5-15 should not surprise us. That's because the digital signals in all three e-mails are identical.

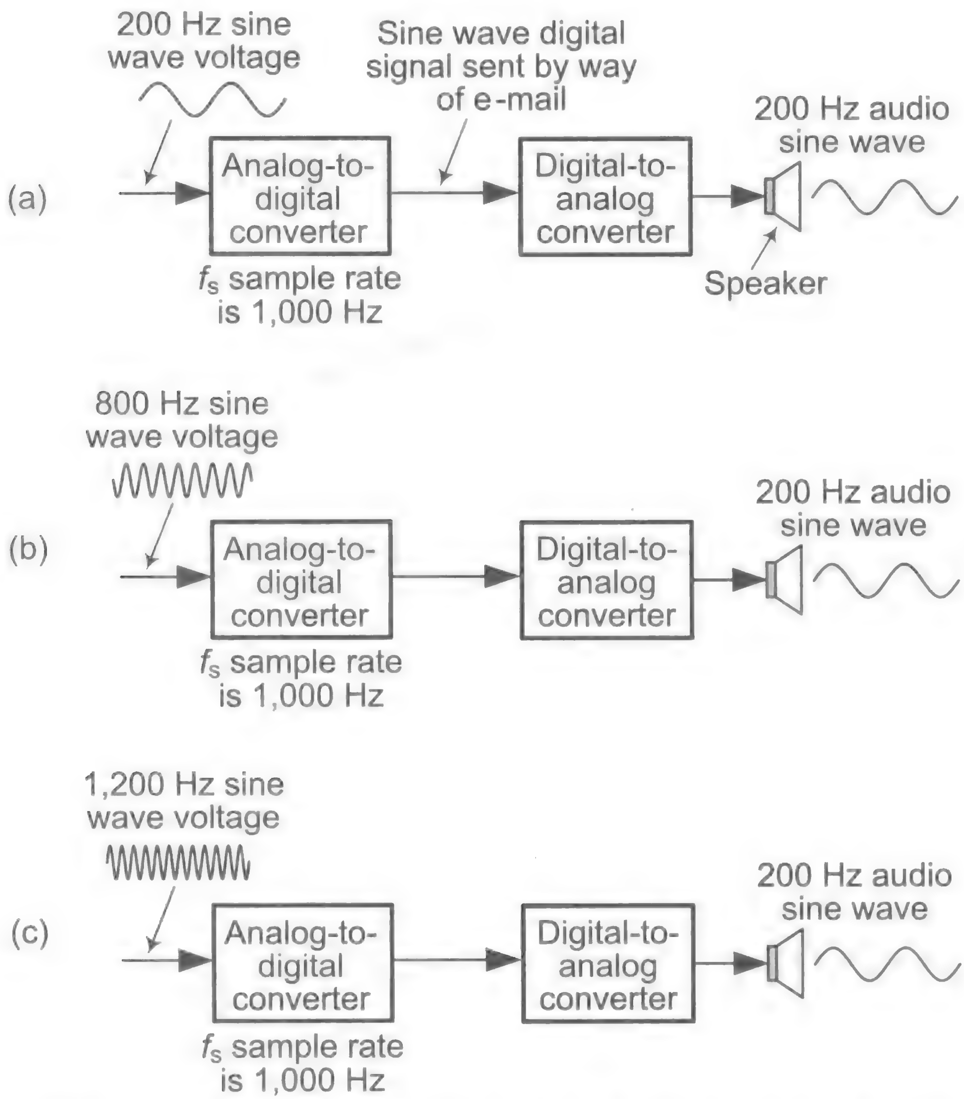


Figure 5-15 Reproducing an analog audio sine wave from a digital signal whose sample rate is 1,000 Hz: (a) original analog sine wave is 200 Hz; (b) original analog sine wave is the alias frequency 800 Hz; (c) original sine wave is the alias frequency 1,200 Hz.

Now we know why 800 and 1,200 Hz sine waves are called **aliases** of a 200 Hz sine wave. *After analog-to-digital conversion, both of these originally high-frequency sine waves will appear to be a low-frequency 200 Hz sine wave*. That's aliasing.

As it turns out, the highest frequency analog sine wave that we can *correctly* represent with digital samples is a sine wave whose frequency is *no greater than* half the sample rate. In our example above, the highest-frequency correctly sampled digital signal sine wave we can e-mail to our friend is a sine wave whose frequency is 1,000/2 = 500 Hz. So here we are, back to the topic we touched upon earlier in this chapter regarding taking photos of the rotating second hand of a clock. That is,

> To correctly represent a continuous (analog) phenomenon whose highest frequency content is f cycles per second (Hz) with a sequence of samples, the f_s sample rate must be greater than two times f.

In the field of digital signal processing, this sample rate restriction is called the **Nyquist sampling criterion**. This constraint is why modern telephone systems that transmit all phone calls as digital signals at a sample rate of f_s = 8,000 Hz must filter all analog audio voice signals so that they contain no analog energy above 8,000/2 = 4,000 Hz, as we discussed in Chapter 3.

THE SPECTRUM OF A DIGITAL SINE WAVE SIGNAL

Now we're ready to answer the question, "What is the spectrum of a digital signal?" We answer that question with an example. Let's say we sampled a 200 Hz analog sine wave at an f_s sample rate of 1,000 Hz as shown in Figure 5-16(a). Figure 5-16(b) shows only the digital signal samples. Next, we compile the samples into a file and e-mail the file to a digital signal processing engineer. We tell the engineer that we sampled an analog sine wave and the digital signal's sample rate is f_s = 1,000 Hz. Being careful not to tell the engineer the original analog signal's frequency, next we ask the engineer to determine the spectrum of the digital signal.

The engineer can perform a mathematical operation on the digital signal samples and determine that the signal has a frequency of one-fifth of the sample rate; that is 1,000/5 = 200 Hz. The engineer can then state that the sampled analog signal that produced the digital signal was a 200 Hz analog sine wave.

But, based on the concept of aliasing we presented in the last section, and an f_s sample rate of 1,000 Hz, the engineer also knows that the original analog signal could also have been an 800, 1,200, 1,800, or 2,200, etc., Hz analog sine wave. Because of the inherent frequency-aliasing behavior of the process of sampling analog signals, the

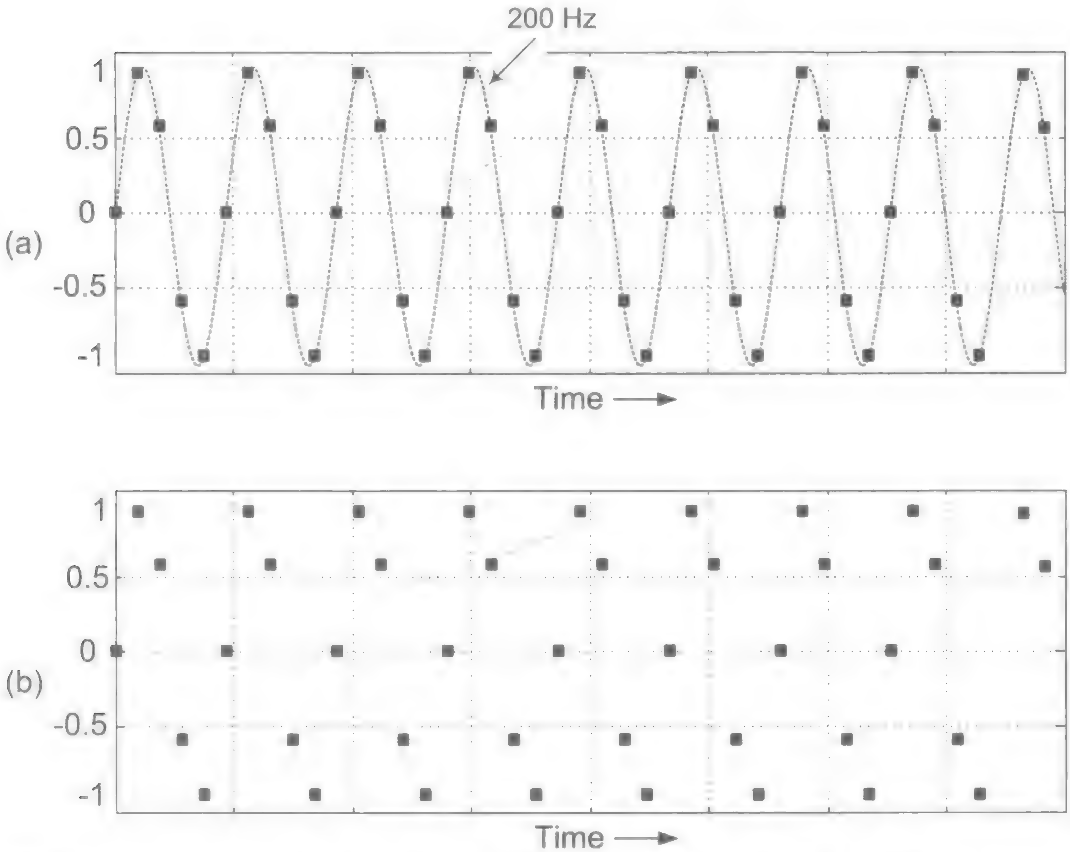


Figure 5-16 Sampling a 200 Hz sine wave at an f_s sample rate of 1,000 Hz: (a) original analog sine wave (dashed curve) and digital signal samples; (b) digital signal samples only.

engineer is faced with a fundamental ambiguity of just what was the frequency of the original analog signal that was sampled to produce the digital signal.

Given this well-known frequency ambiguity, digital signal processing pioneers decided decades ago that the most realistic way to show the spectrum of the Figure 5-16(b) digital signal is the diagram in Figure 5-17. Given the digital signal sample values and the f_s = 1,000 Hz sample rate, the engineer analyzing the digital signal has no choice but to create the Figure 5-17 graphical description, with its never-ending spectral replications. Now you understand why earlier in this chapter we wrote:

> When we convert an analog signal to a digital signal, the spectrum of the digital signal depends on two things: (1) the spectrum of the analog signal and (2) the f_s sample rate of the analog-to-digital conversion process.

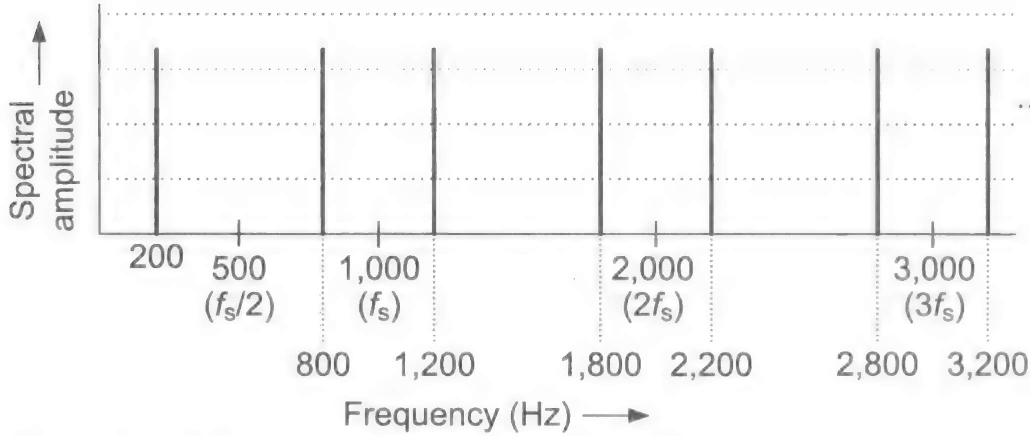


Figure 5-17 The spectrum of the digital signal in Figure 5-16(b).

However, let's say we sent a second e-mail to the engineer stating, "By the way, we *correctly sampled* our analog sine wave. That is, the 1,000 Hz sample rate was greater than twice the frequency of the analog sine wave." At that point, all frequency ambiguity is eliminated for the engineer. He now knows that the analog sine wave signal's frequency was less than $f_s/2$ = 500 Hz, and the only spectral component of the digital signal spectrum in Figure 5-17 that is less than 500 Hz is 200 Hz. Therefore, the sampled analog sine wave had a frequency of 200 Hz, allowing the engineer to graphically describe the spectrum of the digital signal as shown in Figure 5-18.

The point here is that the spectrum of the Figure 5-16(b) digital sine wave signal can be described by either Figure 5-17 or Figure 5-18. Both of these figures are correct, they contain the same amount of information, and, if given one of the figures, we could draw the other figure.

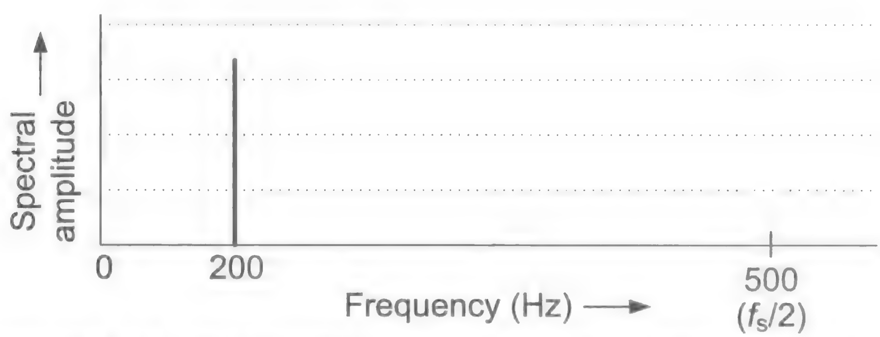


Figure 5-18 The unambiguous spectrum of the digital signal in Figure 5-16(b).

THE SPECTRUM OF A DIGITAL VOICE SIGNAL.........

To enhance our understanding of the spectra of digital signals, let's look at the spectrum of a digital signal that's more complicated than a simple sine wave.

In Chapter 3, we presented an analog voice signal whose spectrum is that shown in Figure 5-19(a). If we apply that analog voice signal to an analog-to-digital converter, whose sample rate was f_s = 8,000 Hz, the spectrum of the digital signal is shown in Figure 5-19(b).

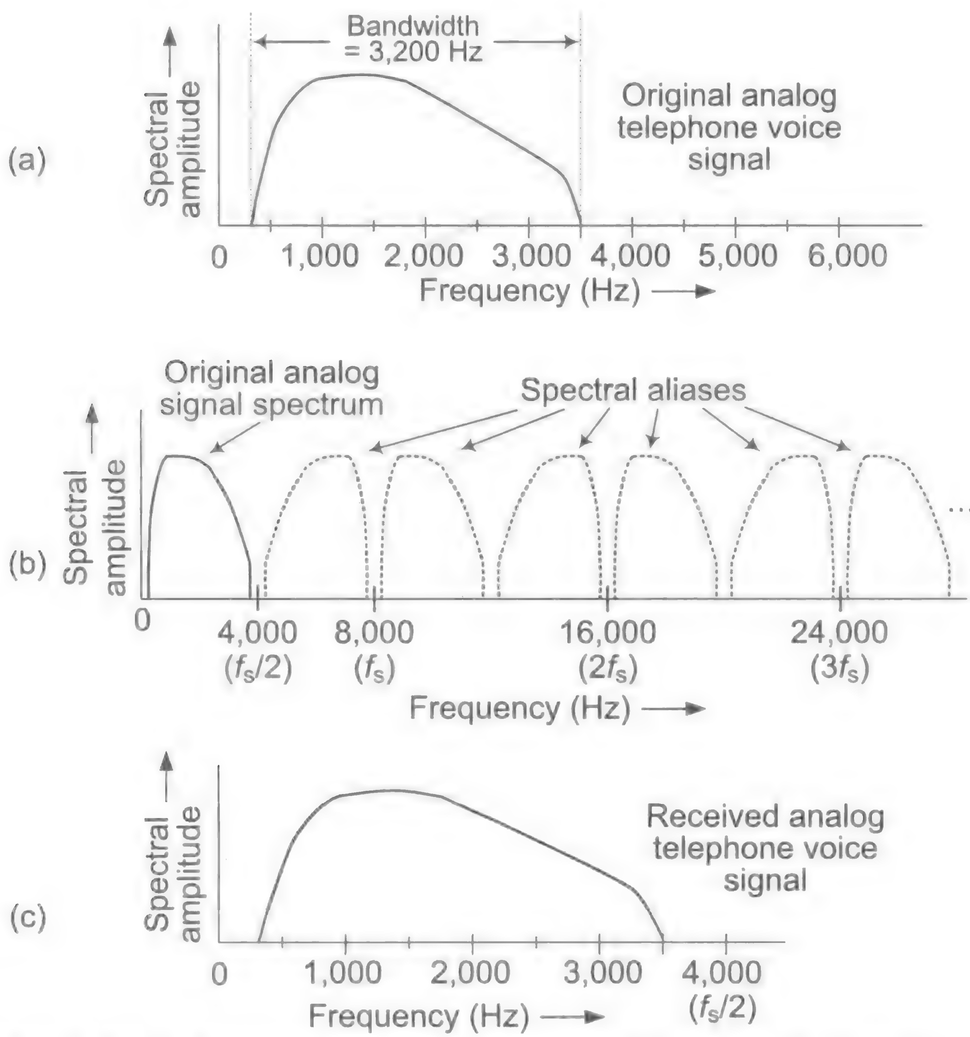


Figure 5-19 Telephone voice signal: (a) original analog signal spectrum; (b) digital signal spectrum showing spectral replications; (c) alternate depiction of the digital signal spectrum.

Because we know the f_s = 8,000 sample rate is greater than twice the highest-frequency spectral component of the original analog voice signal, there is no frequency ambiguity (no overlapping spectral aliases) in Figure 5-19(b). Based on the spectrum of the digital signal, we know the spectrum of the original analog signal is that shown on the left side of Figure 5-19(b). As such, our digital signal's spectrum can also be drawn, with equal validity, as that shown in Figure 5-19(c). Again, both Figure 5-19(b) and Figure 5-19(c) are correct and equivalent depictions of the digital signal's spectrum.

The critical point about this topic of digital voice signals is the fact that the telephone company transmits the digital signal, whose spectrum is shown in Figure 5-19(b), from a switching station near the caller's location to a switching station near the call recipient's location. The switching station near the call recipient's location converts that digital signal back to an analog signal whose spectrum is shown in Figure 5-19(c). The regenerated analog signal is then transmitted to the call recipient's home telephone. Because the spectrum of the analog voice signal received by the call recipient is the same as the spectrum of the analog voice signal originally produced by the caller, as shown in Figure 5-19(a), the call recipient hears an intelligible voice from the loudspeaker of his landline telephone.

THE SPECTRUM OF A DIGITAL MUSIC SIGNAL........

Let's consider audio music signals whose bandwidths are greater than the 3.2 kHz bandwidth of voice signals. Figure 5-20(a) shows the spectrum of a hypothetical analog music signal having a bandwidth of 6 kHz. The highest-frequency spectral content of the analog signal is roughly 6 kHz. If we used an analog-to-digital converter to sample that analog signal at a sample rate of f_s = 8 kHz, the spectrum of the resulting digital signal would be that shown in Figure 5-20(b). Let's assume we've recorded the digital signal on a compact disc (CD).

Because the f_s = 8 kHz sample rate is less than twice the highest-frequency spectral content of the analog signal (that is, less than 12 kHz), a spectral alias overlaps the original signal spectrum as shown by the cross-hatched area on the left side of Figure 5-20(b). In this scenario, we have violated the Nyquist sampling criterion, which results in a digital signal whose low-frequency spectrum is contaminated. As such, if we put that compact disc in a CD player to listen to the music, the audio signal we hear will have the spectrum shown in Figure 5-20(c). The digital signal's overlapped spectral components cause the final analog audio playback signal to have an unpleasant underwater, or gurgling, sound.

Audio engineers who record music on CDs solve the overlapped spectrum problem described above by merely increasing the f_s sample rate of the digital signal obtained from the original analog music signal. Because analog music signals can have frequencies almost as high as 20 kHz, as shown in Figure 5-21(a), the sample

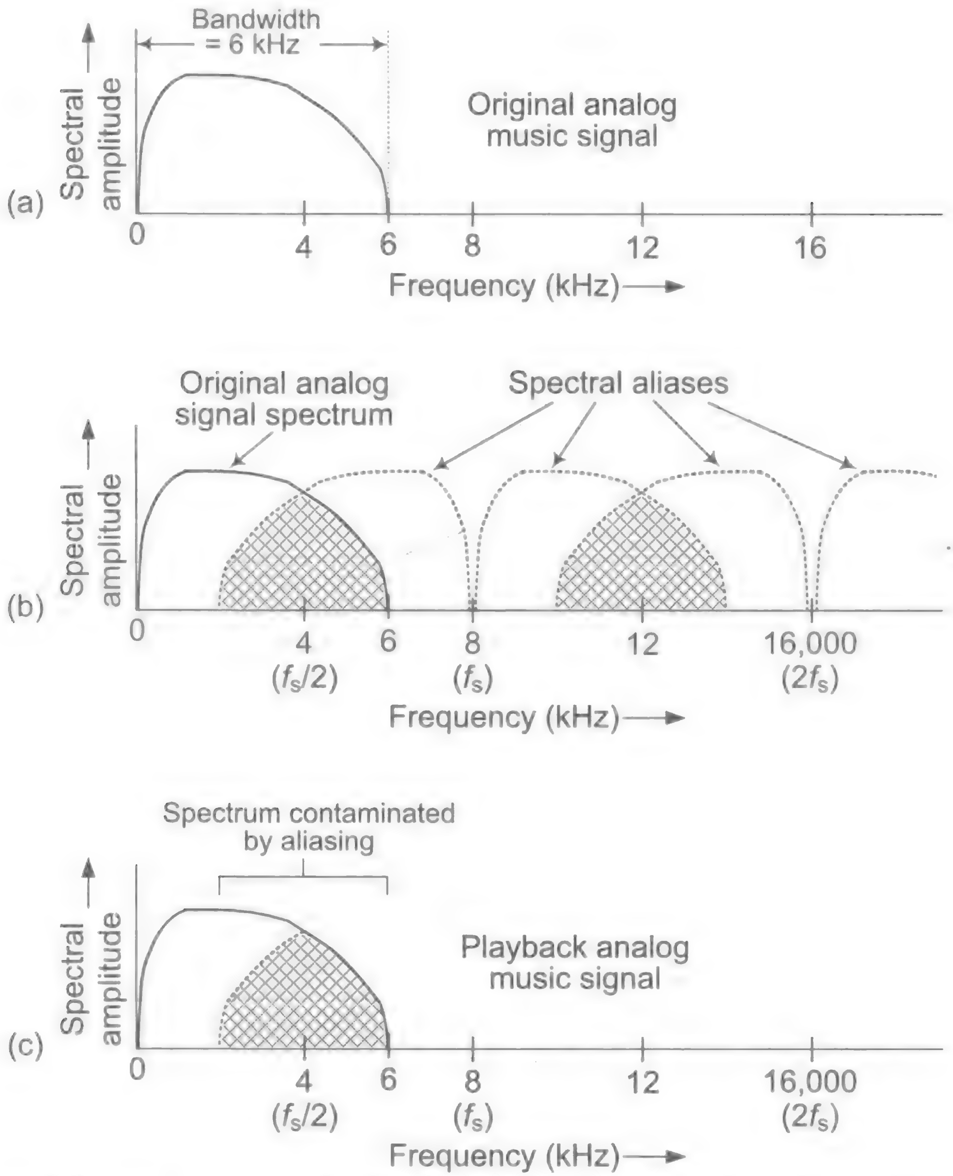


Figure 5-20 Hypothetical analog music signal: (a) original music signal spectrum; (b) digital signal spectrum showing overlapped spectral replications; (c) contaminated spectrum of the analog playback signal from a CD.

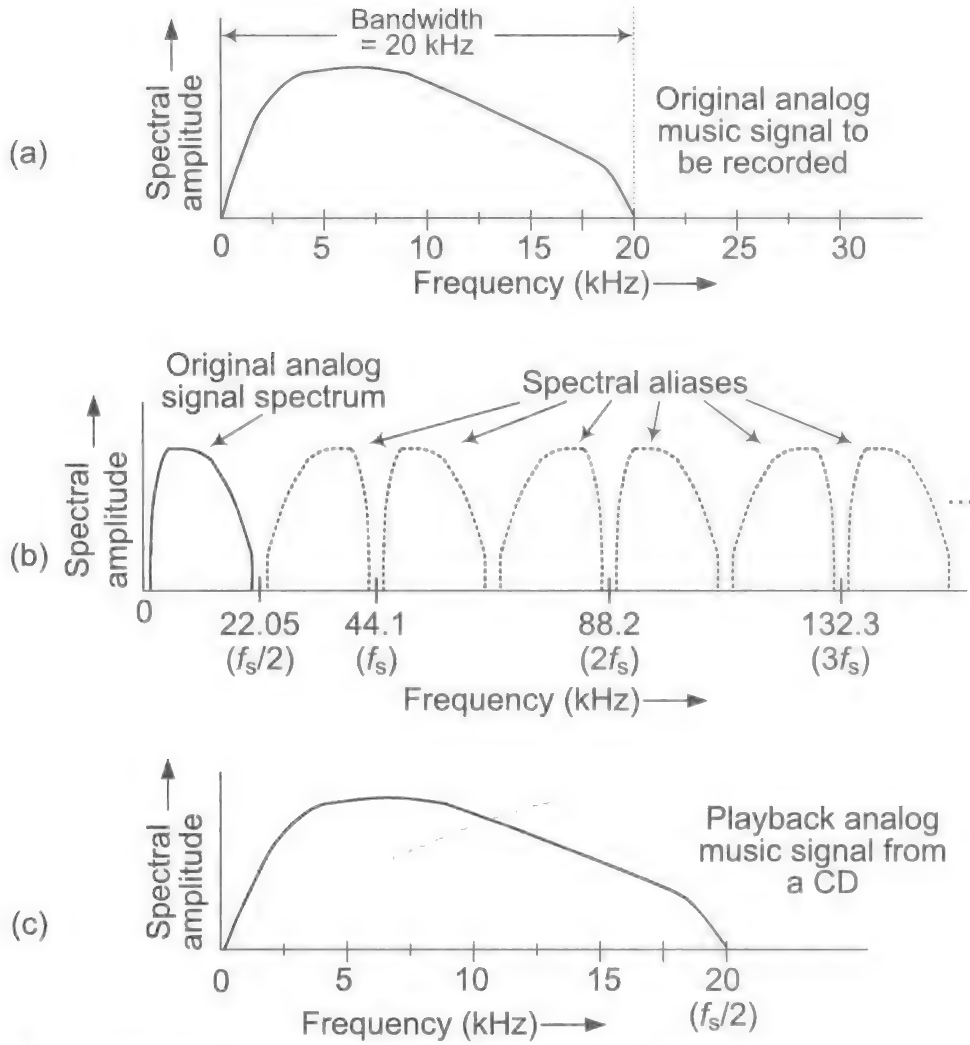


Figure 5-21 Analog music signal: (a) original music signal spectrum; (b) digital signal spectrum showing no overlapped spectral replications; (c) spectrum of the analog playback signal from a CD.

rate of the digital music signals recorded on music CDs must be greater than 40 kHz. For a variety of engineering reasons, the industry standard sample rate for music compact discs is 44.1 kHz as shown in Figure 5-21(b). In that figure, we see that the f_s = 44.1 kHz sample rate results in no spectral overlap and the CD's analog playback signal will be undistorted as shown in Figure 5-21(c).

ANTI-ALIASING FILTERS

In the field of digital signal processing, you're likely to encounter the phrase **anti-aliasing filter**. Such a filter is a hardware device, and we describe its operation by way of a practical example.

Think about a large electric AC power transformer, such as that shown in Figure 5-22, sitting on concrete platform just outside the outer wall of a restaurant or hotel. If you've ever stood next to such a transformer, you may recall how it emits a low-frequency audio hum—a continuous insect-like buzzing. The fluctuating AC voltage inside the transformer generates a fluctuating magnetic field that causes metal plates within the transformer to vibrate. Just as the vibrating cone of a loudspeaker generates sound, the transformer's internal metal plates vibrate and produce a low-level 120 Hz audio hum (or 100 Hz if you live in Europe). Commercial transformer manufacturers do their best to minimize their transformers' audio humming noise. To do so, they must first measure the amplitude (the loudness) of that 120 Hz audio humming signal.

Figure 5-22 Electric AC power transformer. (From Zern Liew/Shutterstock)

So let's say it's our job to measure the amplitude of the audio humming sound in close proximity to a large AC power transformer. Our test setup will be that shown in Figure 5-23(a). In measuring the amplitude of a 120 Hz audio signal, we must satisfy the Nyquist sampling criterion by setting the analog-to-digital converter's f_s sample rate to be greater than two times 120 Hz. So we're free to choose a sample rate of f_s = 300 Hz as shown in Figure 5-23(a).

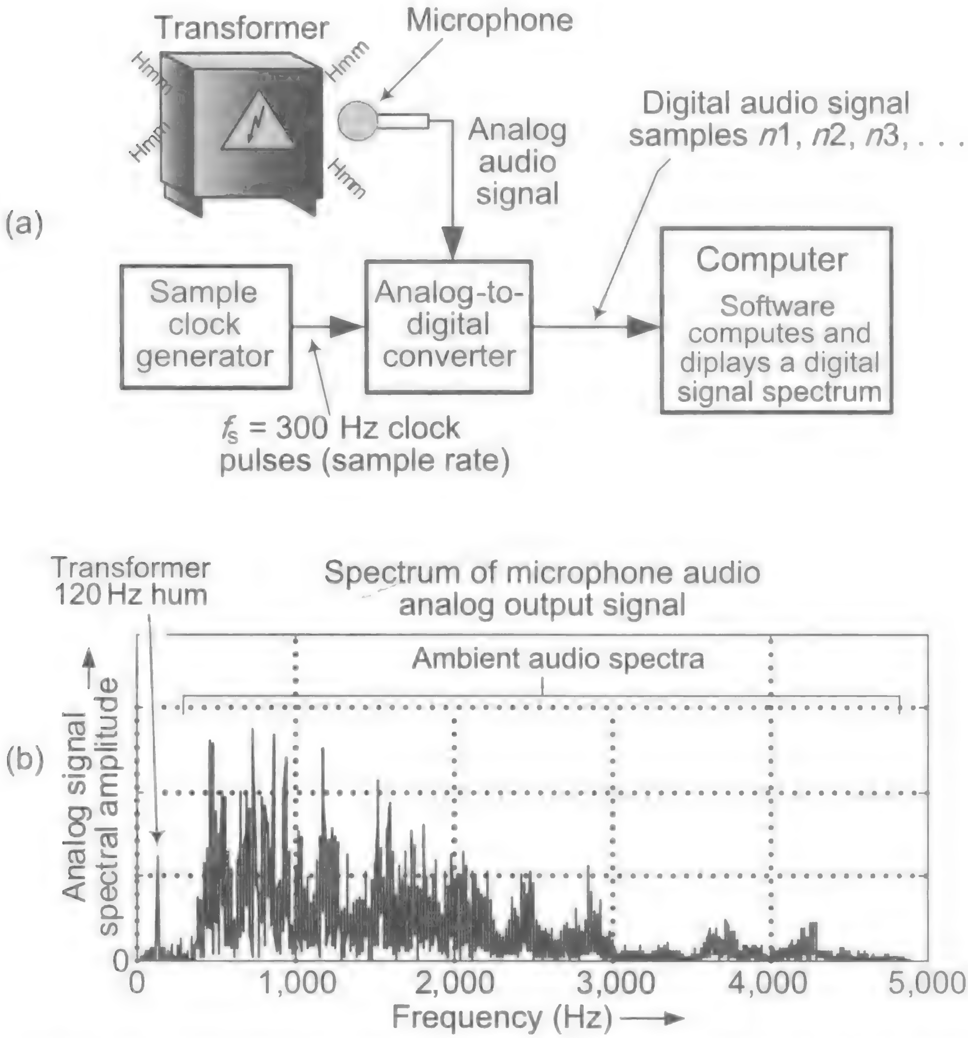


Figure 5-23 *Measuring an AC power transformer's humming 120 Hz audio signal: (a) test equipment setup; (b) spectrum of the microphone's analog output signal.*

Let's assume the spectrum of the microphone's analog audio output signal is as shown in Figure 5-23(b). We see our low-level 120 Hz humming signal's spectral component on the left side of the figure, as well as higher-frequency ambient audio spectral energy caused, for example, by nearby automobile traffic. Again, our job is to measure the amplitude of the 120 Hz signal's spectral component.

Upon computing and displaying the spectrum of our analog-to-digital converter's output signal, shown in Figure 5-24, we encounter a big problem: Due to the inherent behavior of sampling our analog signal, the microphone output signal's high-frequency audio spectral energy aliases down in frequency to appear as the low-frequency spectral energy displayed in Figure 5-24. The 120 Hz spectral amplitude peak value we want to measure is severely contaminated, and essentially obscured, by aliased spectral energy.

The solution to our problem is to eliminate the high-frequency spectral energy in the analog signal applied to our analog-to-digital converter, and we do so with the analog lowpass **anti-aliasing filter** shown in Figure 5-25(a). That analog filter is built to pass low-frequency signals, including our desired 120 Hz signal, and drastically reduces all spectral energy above 120 Hz. Thus, the spectrum of the analog signal applied to the analog-to-digital converter is as shown Figure 5-25(b).

By employing an analog anti-aliasing filter, the ambient high-frequency spectral energy above 120 Hz in Figure 5-25(b), applied to the analog-to-digital converter, is very low in amplitude. So when that unwanted high-frequency, low-amplitude, energy

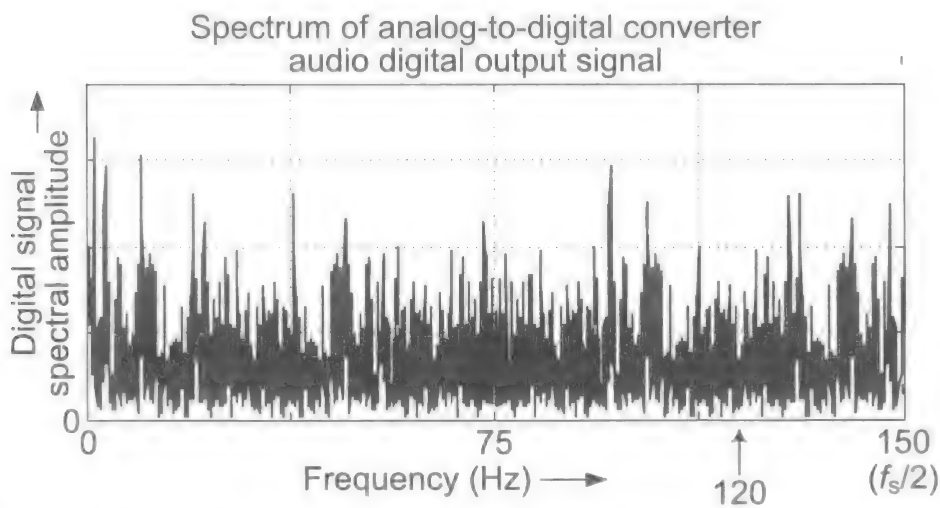


Figure 5-24 Analog-to-digital converter output signal spectrum.

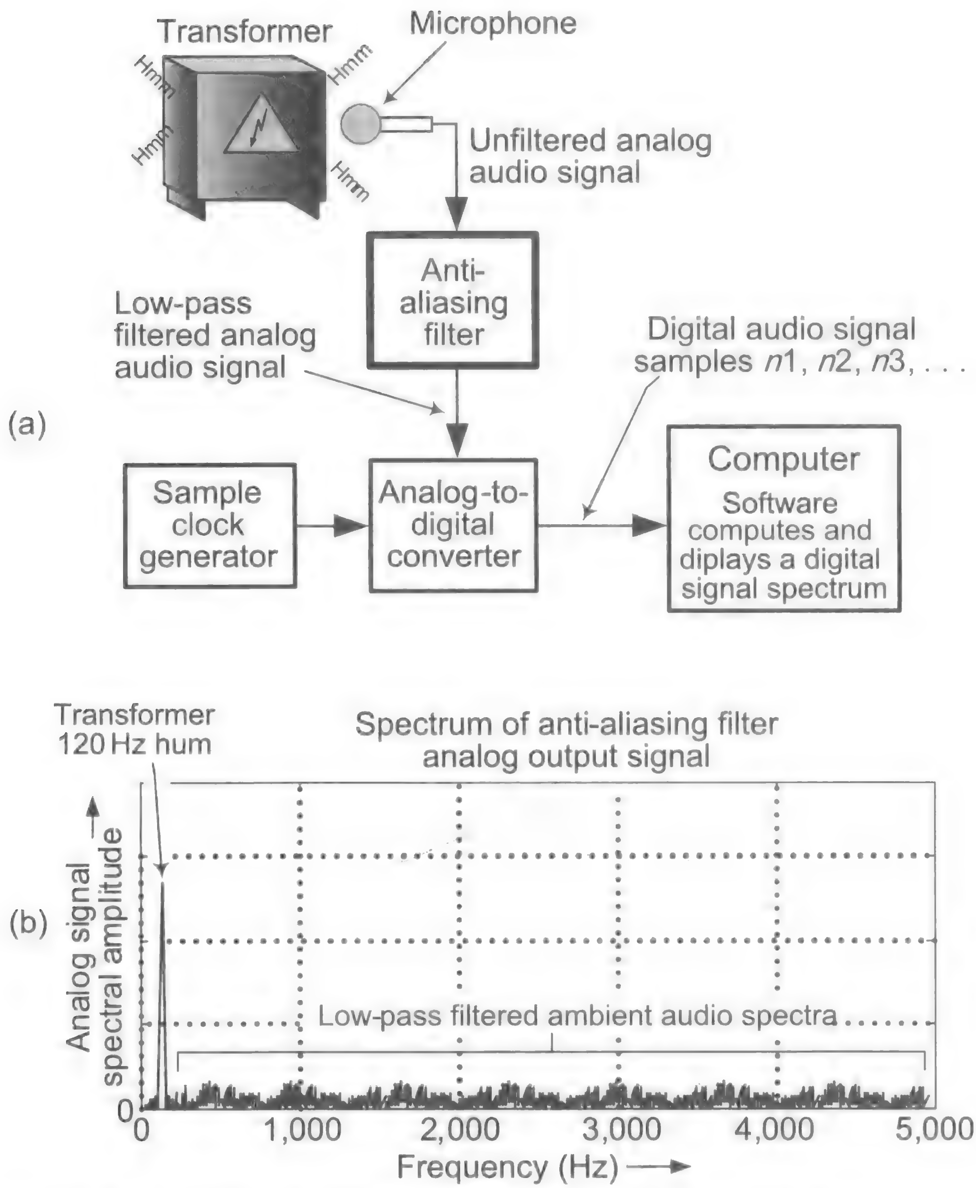


Figure 5-25 Using an anti-aliasing filter to measure a 120 Hz audio signal: (a) new test equipment setup; (b) spectrum of the filter's analog output signal.

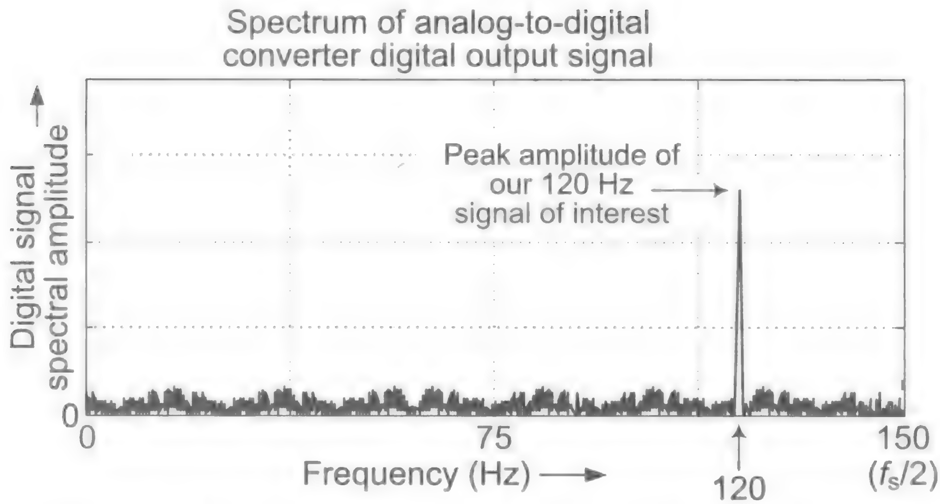


Figure 5-26 *Analog-to-digital converter output signal spectrum.*

aliases down in frequency in our digital signal's spectrum, as shown in Figure 5-26, it does not inhibit our ability to measure the spectral peak amplitude of our desired 120 Hz signal. Using the hardware analog anti-aliasing filter is mandatory for success in this situation.

ANALOG-TO-DIGITAL CONVERTER OUTPUT NUMBERS

An important topic regarding the process of sampling an analog signal is the nature of the numbers produced by an analog-to-digital converter.

When we sample an analog signal, as shown in Figure 5-25(a), the digital signal's sequence of numbers is not in the form of the decimal numbers that we're so familiar with in our daily lives. The digital signal's sequence of numbers, *n*1, *n*2, *n*3, . . . , is in the form of what we call **binary numbers**. The interesting topics of binary numbers and why we use them are discussed in Chapter 9.

WHAT YOU SHOULD REMEMBER

The concepts you should remember from this chapter are:

- To correctly represent a continuous (analog) phenomenon whose highest frequency content is f cycles per second (Hz) with a sequence of samples, the sample rate must be greater than two times f (the Nyquist sampling criterion).
- After analog-to-digital conversion using a sample rate of f_s Hz, any sampled analog sine wave whose frequency is greater than $f_s/2$ Hz will always be translated down in frequency to be in the range of zero Hz to $f_s/2$ Hz.
- Any digital sine wave signal (a sequence of discrete samples), such as that in Figure 5-13, represents a sampled version of an infinite number of high-frequency analog sine waves.
- In many real-world applications of sampling analog signals, hardware anti-aliasing filters are used to eliminate the unwanted high-frequency content of an analog signal applied to an analog-to-digital converter.

6 How We Compute Digital Signal Spectra

COMPUTING DIGITAL SPECTRA

One of the most important aspects of digital signal processing is the computation, and display on a computer screen, of the spectral amplitude of a digital signal as shown in Figure 6-1. The ability to examine the spectrum of a digital signal is as crucial to a signal processing engineer as a microscope is to a medical researcher.

Today, there are two primary methods available to compute digital signal spectra, and those two procedures have the fancy names of the **discrete Fourier transform** and the **fast Fourier transform**. This chapter briefly introduces those two functionally equivalent spectral computational methods, and ends with an example of computing a digital signal spectrum for the interested reader.

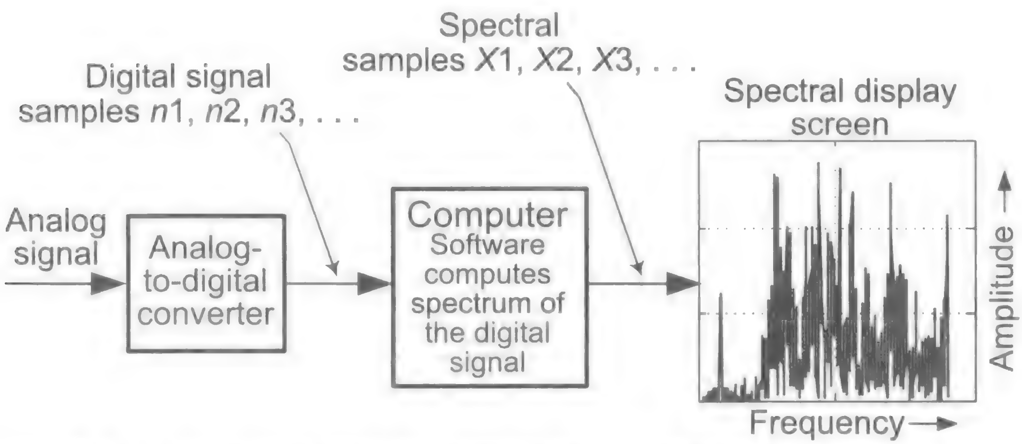


Figure 6-1 Computing and displaying the spectrum of a digital signal.

The Discrete Fourier Transform

The **discrete Fourier transform** is a mathematical process in which an input sequence of digital signal samples is correlated with an array of digital sine and cosine sequences of different frequencies to compute a new sequence of samples that represents the spectral content of the original digital signal. Whew! That was a mouthful! Let's just say that the discrete Fourier transform (popularly known as the **DFT**) is a mathematical process performed on digital signal sequences to compute their spectra, and those spectral results can then be plotted on a computer screen.

Although basically a set of arithmetic comparisons or correlations, the discrete Fourier transform is far too mathematically complicated to describe its inner algebraic details in any meaningful way here. However, we must mention one of the discrete Fourier transform's important characteristics. Performing a discrete Fourier transform requires an astonishingly large number of arithmetic computations. For example, let's say we want to compute the spectrum of just a one-second interval of a digital voice signal sampled at a rate of f_s = 8,000 Hz. Using the discrete Fourier transform to compute the digital voice signal's spectrum requires us to perform over 128 million addition operations and over 256 million multiplication operations to compute the spectral $X1$, $X2$, $X3$, . . . sample values as depicted in Figure 6-1. To call this "some serious number crunching" is putting it far too mildly. And the computational workload to obtain the spectrum of longer time-duration signals, sampled at higher f_s sample rates, becomes truly astronomical. Fortunately, thanks to the brilliant work of dead mathematicians and the use of modern digital computers, today there is a more efficient way to compute digital signal spectra.

The Fast Fourier Transform

The **fast Fourier transform**, developed in the United States in the mid-1960s, is an alternate mathematical technique for computing the spectra of digital signals. In fact, the fast Fourier transform (popularly known as the **FFT**) computes identical results to those computed by the discrete Fourier transform. However, the number of arithmetic computations needed by the fast Fourier transform compared to the discrete Fourier transform is dramatically reduced.

As an example of the fast Fourier transform's computational efficiency, recall the scenario above where we wanted to compute the spectrum of a one-second interval of a digital voice signal sampled at a rate of f_s = 8,000 Hz. Doing this using the fast Fourier transform requires roughly 100,000 addition operations and 200,000 multiplication operations. That's a computation reduction by a factor of 1,000 compared to the computations required by the discrete Fourier transform. What we're saying is that for *each* multiplication operation required by the fast Fourier transform, the discrete

Fourier transform requires 1,000 multiplications! Consequently, the fast Fourier transform is the spectral computation method of choice among modern signal processing engineers.

By the Way

The Fourier transform is named after the 19th-century French mathematician and scientist, Jean Baptiste Joseph Fourier. (The name Fourier is pronounced 'for-YAY, like obey.) A friend of Napoléon Bonaparte, in 1822 Fourier was the first person to propose that periodic waveforms can be described as the sum of various sinusoidal waveforms.

A SPECTRAL COMPUTATION EXAMPLE ....

This section provides, for those courageous readers willing to continue, a simple example of how the spectrum of a digital signal is computed. In Chapter 3, we discussed a square wave-like analog signal that was the sum of a 2 Hz sine wave plus lower-amplitude 6 Hz and 10 Hz sine waves as shown in Figure 6-2(b). A digital signal version of the composite signal is given in Figure 6-2(c).

The Computations

The goal in our example is to compute the spectrum of the Figure 6-2(c) digital signal. Conceptually, both the discrete Fourier transform and the fast Fourier transform compute the spectrum of the Figure 6-2(c) digital signal sequence as follows:

1. The digital signal sequence has 40 samples; $n1$, $n2$, $n3$, . . . , $n40$, as shown in Figure 6-3(a). First, we create a single-cycle sine wave sequence having 40 samples as shown in Figure 6-3(b).
2. Multiply each sample in the digital signal by its corresponding sample in the single-cycle sine wave sequence, resulting in 40 products. That is, the first product is $P1 = n1 \times s1$. (The "x" means times, a multiplication.) The second product is $P2 = n2 \times s2$, the third product is $P3 = n3 \times s3$, and so on up to the 40th product $P40 = n40 \times s40$.
3. Then, we sum the 40 products to obtain our first spectral sample value $X1$. That is, $X1 = P1 \times P2 + P3 + \ldots + P40$. Because some of the products are positive and some of the products are negative, for our example those products sum to exactly zero. So $X1 = 0$.

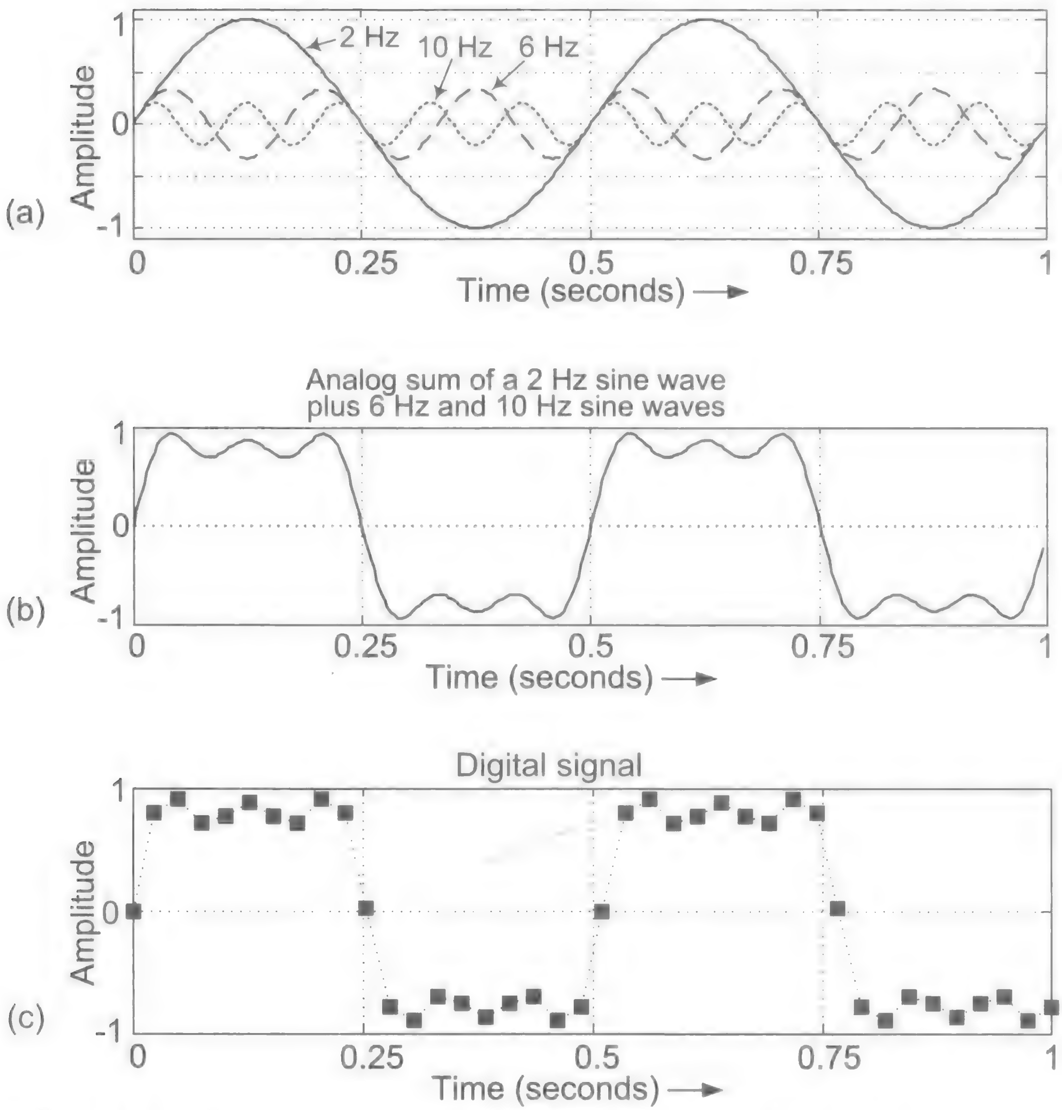


Figure 6-2 A square wave-like signal: (a) individual analog 2 Hz, 6 Hz, and 10 Hz sine waves; (b) analog sum of the sine waves; (c) a digital signal version of the sum of the sine waves sampled at a rate of 40 samples per second (f_s = 40 Hz).

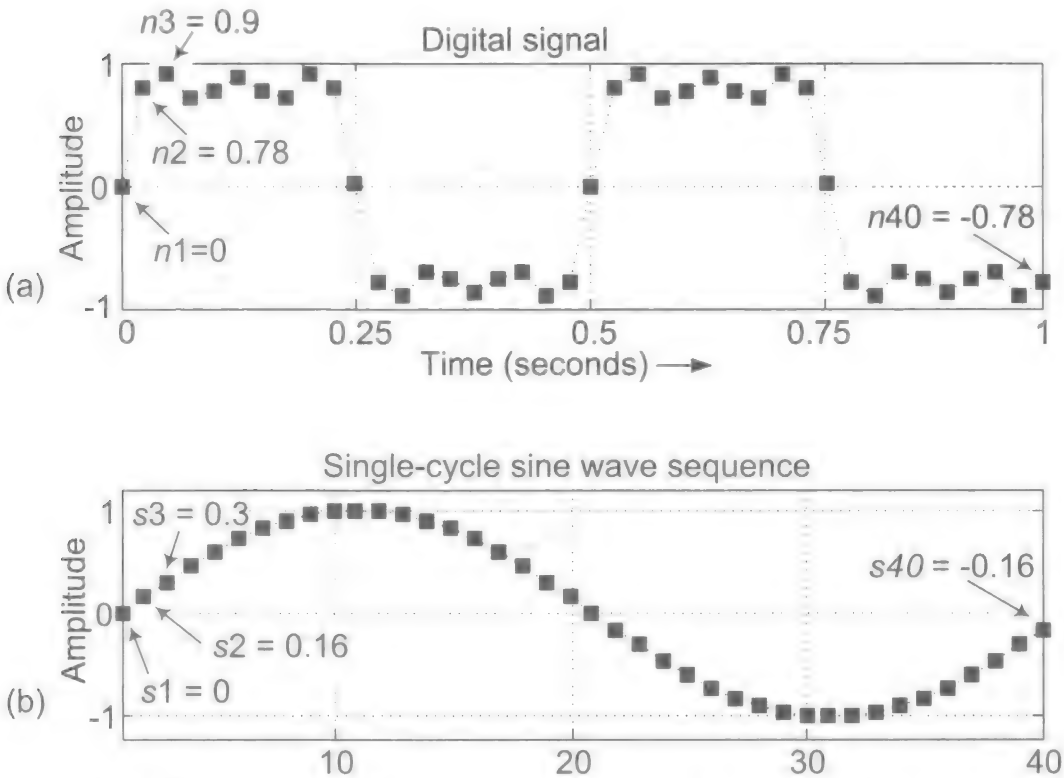


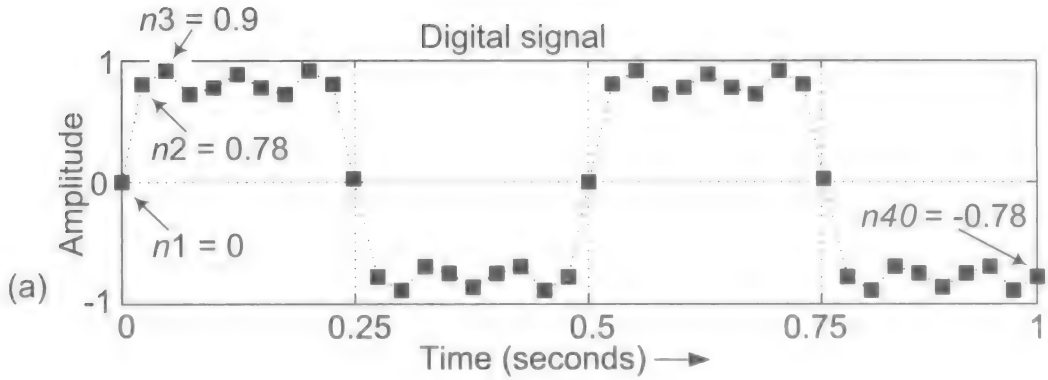
Figure 6-3 Computing the first spectral sample value $X1$: (a) the digital signal sequence; (b) a single-cycle sine wave sequence.

4. Next, create a two-cycle sine wave sequence, having 40 samples, as shown in Figure 6-4(b).

5. Multiply each sample in the digital signal by its corresponding sample in the two-cycle sine wave sequence, resulting in 40 products. Again, the first product is $P1 = n1 \times s1$. The second product is $P2 = n2 \times s2$, the third product is $P3 = n3 \times s3$, and so on up to the 40th product $P40 = n40 \times s40$.

6. Sum the 40 products to obtain our second spectral sample value. That is, $X2 = P1 + P2 + P3 + \ldots + P40$. For our example, those products sum to 20. So $X2 = 20$.

7. Next, create a three-cycle sine wave sequence, having 40 samples, as shown in Figure 6-5(b).



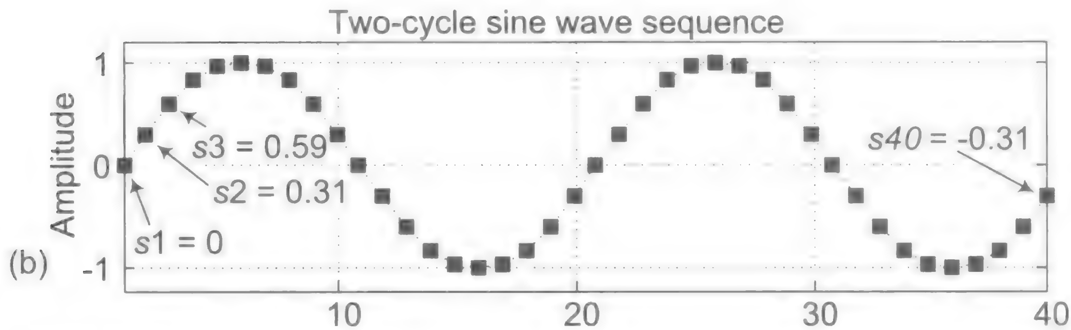


Figure 6-4 Computing the second spectral sample value $X2$: (a) the digital signal sequence; (b) a two-cycle sine wave sequence.

8. As before, multiply each sample in the digital signal by its corresponding sample in the three-cycle sine wave sequence, again resulting in 40 products.
9. Sum the 40 products to obtain our third spectral sample value $X3 = P1 + P2 + P3 + \ldots + P40$. For our example, those products sum to zero. So $X3 = 0$.
10. Repeat steps 7 through 9 another 17 times, increasing the number of cycles in the sine wave sequence by one each time, until we have computed 20 spectral samples values, $X1$ to $X20$.
11. Plot the 20 spectral samples values on a graph with frequency as the horizontal axis.

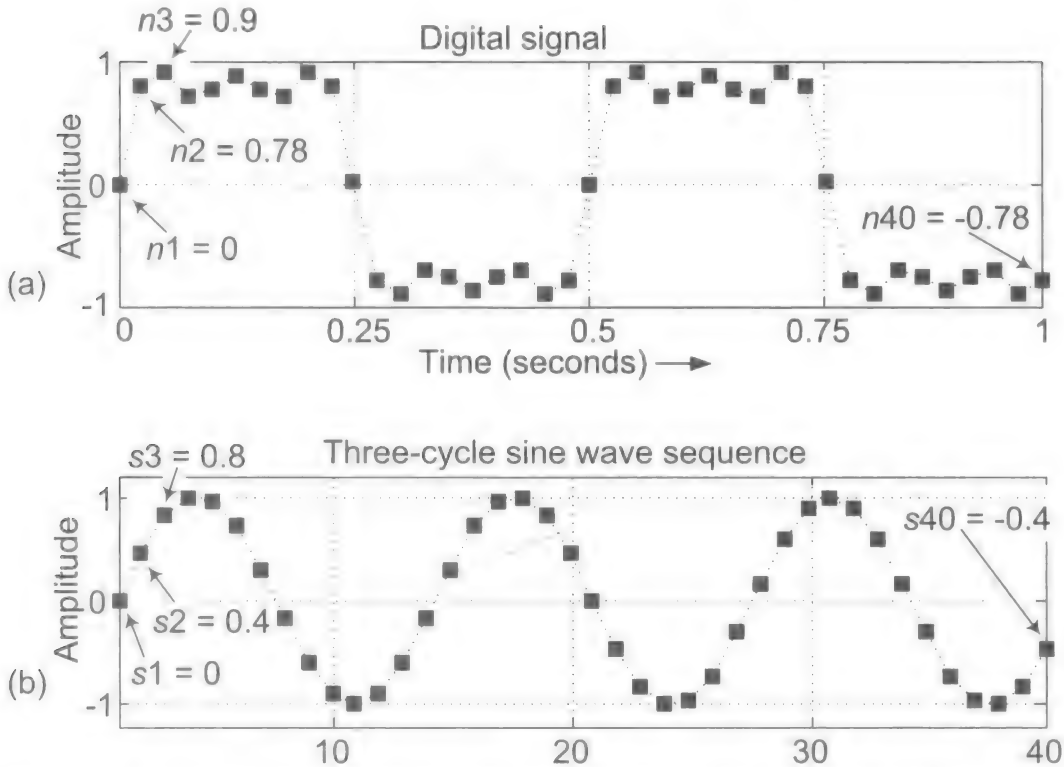


Figure 6-5 Computing the third spectral sample value $X3$: (a) the digital signal sequence; (b) a three-cycle sine wave sequence.

Performing the steps above for our square wave-like digital signal gives us the 20 spectral sample values listed in Table 6.1. And, finally, plotting those spectral sample values produces our desired spectral plot shown in Figure 6-6. (Figures 6-6(a) and 6-6(b) show two popular plotting methods. One plot uses dots to represent the spectral samples, and the other plot connects the spectral sample values with solid lines and deletes the dots.) In that figure, we see that the digital signal contained a high-amplitude 2 Hz spectral component, a lower-amplitude 6 Hz spectral component, and a slightly lower-amplitude 10 Hz spectral component. Upon reviewing Figure 6-2(a), we see that our Figure 6-6 spectral results are correct.

Table 6.1 Spectral Sample Values

Spectral Sample Values	
$X1 = 0$	$X11 = 0$
$X2 = 20$	$X12 = 0$
$X3 = 0$	$X13 = 0$
$X4 = 0$	$X14 = 0$
$X5 = 0$	$X15 = 0$
$X6 = 6.66$	$X16 = 0$
$X7 = 0$	$X17 = 0$
$X8 = 0$	$X18 = 0$
$X9 = 0$	$X19 = 0$
$X10 = 4$	$X20 = 0$

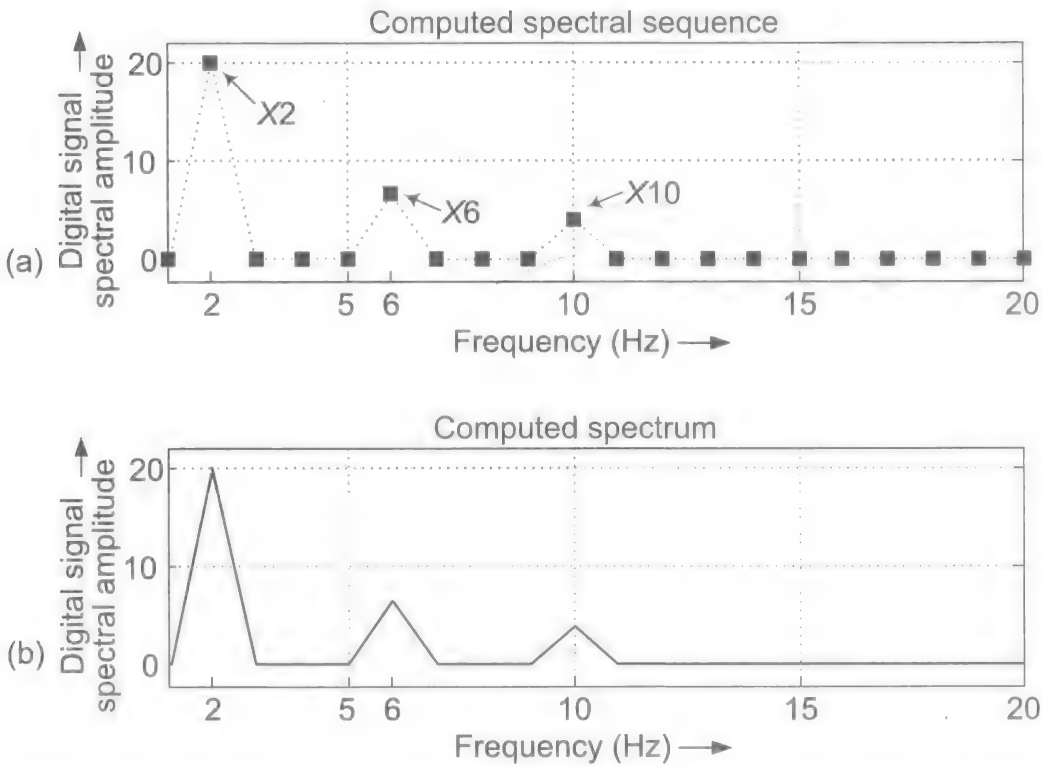


Figure 6-6 Spectrum of the square wave-like digital signal. Two plotting formats: (a) using dots to represent the spectral samples; (b) solid lines connecting the spectral sample values and dots removed.

What the Computations Mean

For the example above, in Figure 6-3 when we multiplied each sample in the digital signal by its corresponding sample in the single-cycle sine wave sequence and summed the 40 products, we arrived at a value of $X1 = 0$. That *summing of products* operation gives us a single numerical value, $X1 = 0$, indicating how well the single-cycle sine wave sequence compared with the square wave-like digital signal. The $X1 = 0$ value tells us our digital signal compares *very* poorly with the 1 Hz sine wave sequence. Stated differently, the $X1 = 0$ value reveals that our digital signal contains a zero amount of 1 Hz spectral energy.

However, in Figure 6-4 we multiplied each sample in the digital signal by its corresponding sample in the two-cycle sine wave sequence and summed the 40 products to yield a value of $X2 = 20$. The $X2 = 20$ value tells us our digital signal compares *very* well with the 2 Hz sine wave sequence. (The peaks and valleys of the two-cycle sine wave sequence are very well aligned with the positive and negative values of our square wave-like digital signal.) That is, the $X2 = 20$ value reveals that our digital signal contains a large amount of 2 Hz spectral energy.

As we continued our summing of products operations, we found that our numerical comparisons were always zero-valued except when the sine wave sequences had frequencies of 2, 6, and 10 Hz. Therefore, our square wave-like digital signal contained varying amounts of spectral energy at the frequencies of 2, 6, and 10 Hz.

By the Way

Mathematicians have a name for our summation of products operation, and you've heard that name before. It's called *correlation*. In Chapter 7, Wavelets, we'll read about comparisons that involve waveforms other than sine waves. Stay tuned.

A SPECTRAL ANALYSIS EXAMPLE

Let's consider a real-world spectral analysis example. Think about a large electric motor, shown in Figure 6-7(a), located on the factory floor of a manufacturing company. (By large, we mean, say, the size of a beer keg.) Perhaps the motor drives a heavy-duty hydraulic pump, or maybe a large conveyor belt. In any case, let's assume the motor's drive gear turns a larger gear as shown in Figure 6-7(b).

When the motor, and any attached equipment, is first installed, the teeth of the gears mesh very well as shown in Figure 6-7(b). But over time, the drive gear becomes worn as shown in Figure 6-7(c). And when the teeth of the drive gear become badly worn, an unexpected equipment failure can occur—and factory engineers don't like unexpected equipment failures.

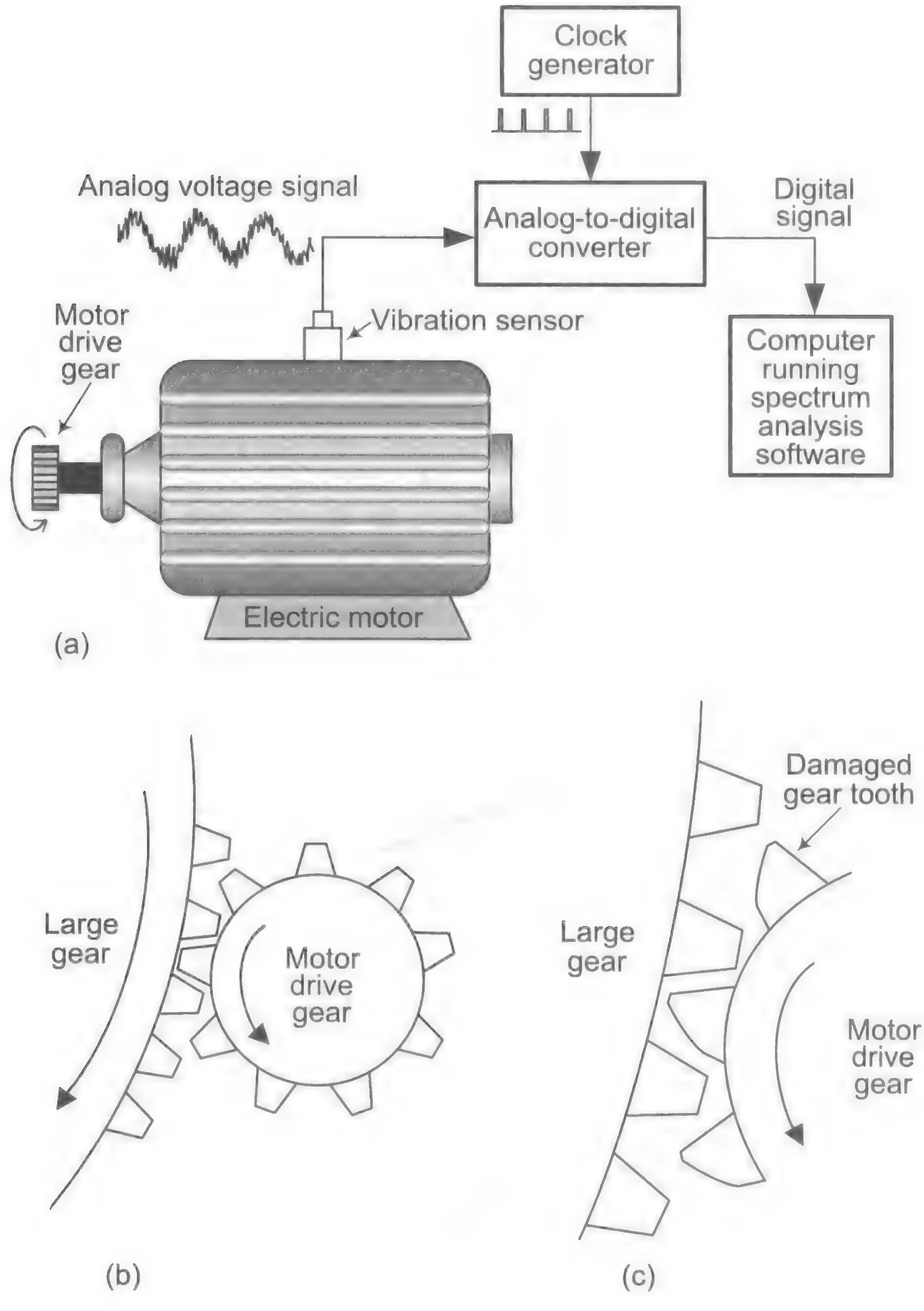


Figure 6-7 Electric motor and gears: (a) motor and vibration test setup; (b) new motor drive gear; (c) worn motor drive gear.

Those clever engineers are able to avoid unexpected electric motor failures by performing spectrum analysis on the vibration of their electric motors. Here's how: As it turns out, when a new motor, and any connected mechanical equipment, is first installed in a factory, engineers attach a vibration sensor to the motor's case as shown in Figure 6-7(a). (Just as a microphone converts a sound signal, which consists of fluctuations in air pressure, into a voltage signal, the vibration sensor converts mechanical vibrations into an electrical voltage.) By means of an analog-to-digital converter, the analog vibration signal is converted to a digital signal, which is then passed to a computer. The computer uses fast Fourier transform software to compute and display the vibrational spectrum of a new factory installation. If the motor is rotating at 20 revolutions per second, and the motor's drive gear has 12 teeth, the vibration spectrum will contain a $20 \times 12 \times 240$ Hz spectral component (in addition to the 20 Hz component from the motor itself) as shown in Figure 6-8(a).

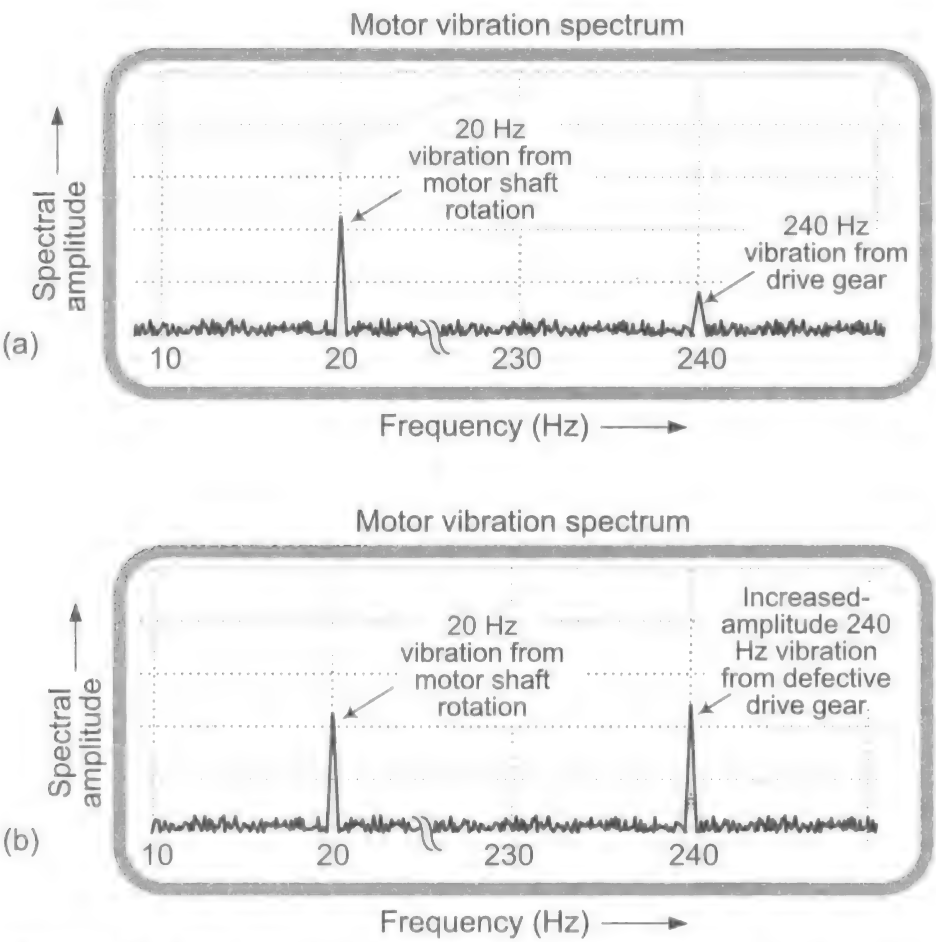


Figure 6-8 Motor vibration spectra: (a) new motor vibration spectrum; (b) worn-gear motor vibration spectrum.

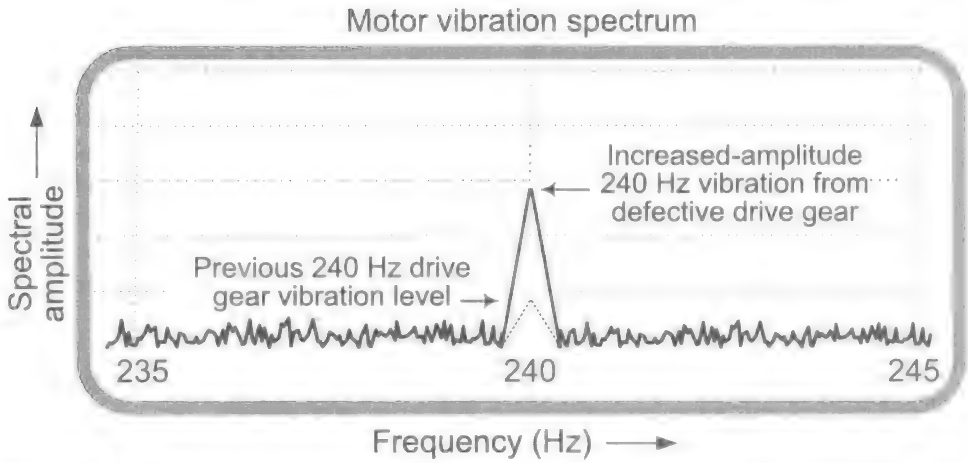


Figure 6-9 Worn-gear motor vibration spectrum detail.

Now when the teeth of the motor's drive gear become worn, it causes increased motor-case vibration. So some months after installation, the factory engineers repeat their motor vibration spectral test. Let's say the new spectrum display is that shown in Figure 6-8(b). Seeing an increased amplitude of the 240 Hz spectral component, the engineers zoom in on the spectral display at the 240 Hz frequency as shown in Figure 6-9. From that display, the engineers know that the drive gear's teeth are becoming worn. This way, the engineers can schedule corrective maintenance that is the least disruptive to the factory's production process.

WHAT YOU SHOULD REMEMBER ....

The concepts you should remember from this chapter are:

- The discrete Fourier transform (DFT) and the fast Fourier transform (FFT) are functionally equivalent mathematical methods for computing the spectra of digital signals.
- The discrete Fourier transform requires a very large number of arithmetic computations.
- The fast Fourier transform requires relatively few arithmetic computations, and is now the most popular method for computing the spectra of digital signals.

7 Wavelets

There is a discipline in the field of digital signal processing called wavelets. As we'll explain here, wavelet transforms are similar to Fourier transforms except that with wavelets, we can find both the frequency *and* the time of interesting signal characteristics. Wavelet processing is used extensively in signal and image processing, medicine, finance, radar, sonar, geology, and many other varied fields. Wavelets are usually presented in mathematical formulae, but can actually be understood in terms of simple comparisons with the signal data we are analyzing.

To provide some background, we first look at the fast Fourier transform (FFT). That transform can be thought of as a series of comparisons with your data, which for now we will call a signal for consistency. Signals whose frequency content does not change over time can be processed with ordinary FFT methods.

Real-world signals, however, often have frequencies that change over time or have pulses, anomalies, or other *events* at certain specific times. This type of signal can tell us where something is located on the planet, the health of a human heart, the position and velocity of a blip on a radar screen, stock market behavior, or the location of underground oil deposits. For these signals, wavelet analysis has the capability to show not only the spectral content of a signal, but also to indicate when in time those spectral components exist. That is a very powerful signal analysis capability. We now demonstrate both the fast Fourier and wavelet transforms of a simple pulse signal.

THE FAST FOURIER TRANSFORM (FFT)— A QUICK REVIEW

We start with a point-by-point comparison of the pulse signal (D) with a high-frequency sinusoid wave of constant frequency (A) as shown in Figure 7-1. We obtain a single *goodness* value from this comparison (a correlation value), which indicates how much of that particular sinusoid wave is found in our pulse signal.

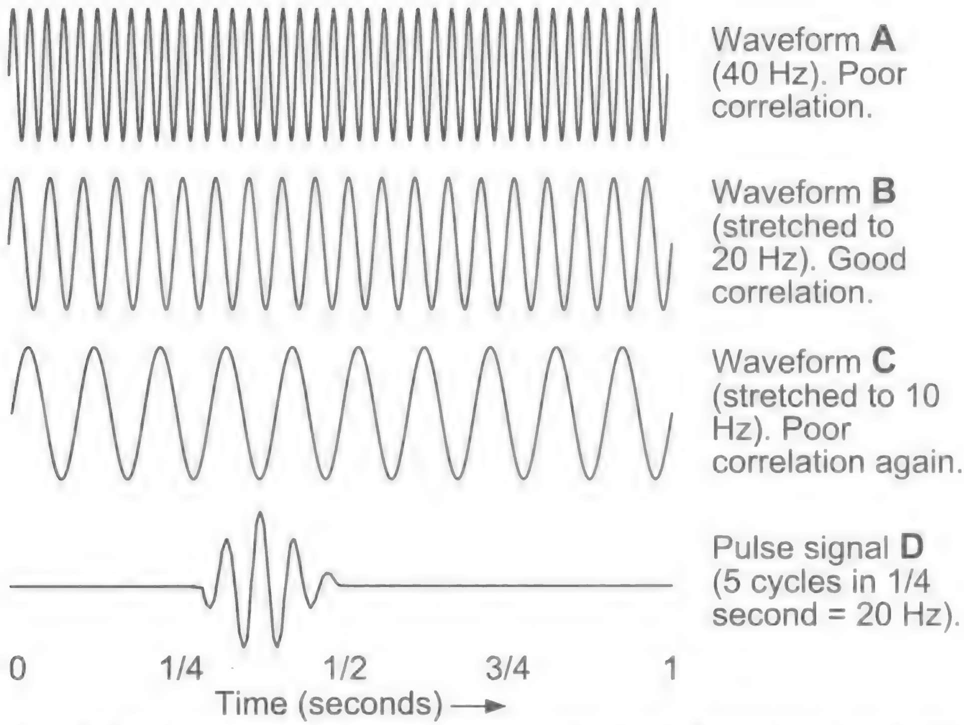


Figure 7-1 Fast Fourier transform (FFT) comparison of example waveforms.

We can observe that the pulse D has 5 cycles in one-quarter of a second. This means it has a frequency of 20 cycles in one second or 20 Hz. The comparison sinusoid, A, has twice the frequency or 40 Hz. Even in the area where the D signal is nonzero (the pulse), the comparison is not very good.

By lowering the frequency of A from 40 to 20 Hz (waveform B), we are effectively *stretching* the sinusoid (A) by 2 so it has only 20 cycles in 1 second. We compare point-by-point again over the 1-second interval with the pulse D. This correlation gives us another value that indicates how much of this lower frequency sinusoid (now the same frequency as our pulse) is contained in our pulse signal. This time, the correlation of the pulse with the comparison 20 Hz sinusoid is very good. The peaks and valleys of B and the pulse portion of D align (or can be easily shifted to align) and thus we have a large correlation value.

Figure 7-1 shows us one more comparison of our original sinusoid A stretched by 4 so it has only 10 cycles in the 1-second interval, C. The comparison of the 10 Hz sine wave with pulse signal D is again poor. We could continue stretching until the comparison sine wave becomes a straight line having zero frequency, but all those comparisons will be increasingly poor.

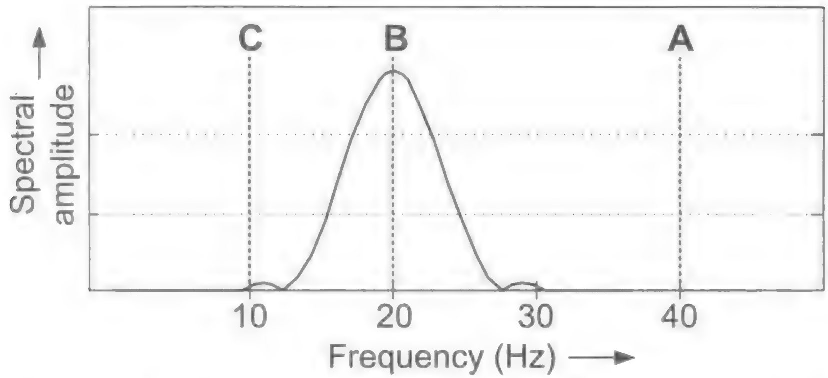


Figure 7-2 FFT spectral plot of Figure 7-1's D pulse signal, with the three sinusoids indicated.

A fast Fourier transform (FFT) compares many stretched sinusoids to the pulse rather than just the 3 shown in Figure 7-1. The best correlation is found when the sinusoid frequency best matches the frequency of the pulse. Figure 7-2 shows the first part of an actual FFT of our pulse signal D. The locations of our sample comparison sinusoids A, B, and C are indicated. Notice that the FFT correctly tells us that the pulse has primarily a frequency of 20 Hz, but does *not* tell us where the pulse is located in time!

THE CONTINUOUS WAVELET TRANSFORM (CWT)

Wavelets are exciting because they, too, are comparisons, but instead of correlating with various stretched, infinite-length, unchanging sinusoids, they use smaller or shorter waveforms (wave-lets) that can start and stop where we wish.

Using what is called the *continuous wavelet transform* (CWT), by stretching and shifting the wavelet numerous times we obtain numerous correlations. If our signal under analysis has some interesting events embedded in it, we will get the best correlation when the stretched wavelet is similar in frequency to the event and is shifted to line up in time with the event. Knowing the amounts of stretching and time shifting, which produce large comparison result values, we can determine both the frequency, and time of occurrence, of interesting events within a signal.

Figure 7-3 demonstrates the process. Instead of sinusoids for our comparisons, we will use wavelets. Waveform A shows what is called a *Daubechies 20* wavelet about one-eighth of a second long that starts at the beginning ($t = 0$) and effectively ends well before one-quarter of second. The zero values are extended to the full 1 second. The point-by-point comparison with our pulse signal D will be very poor and we will obtain a very small correlation value.

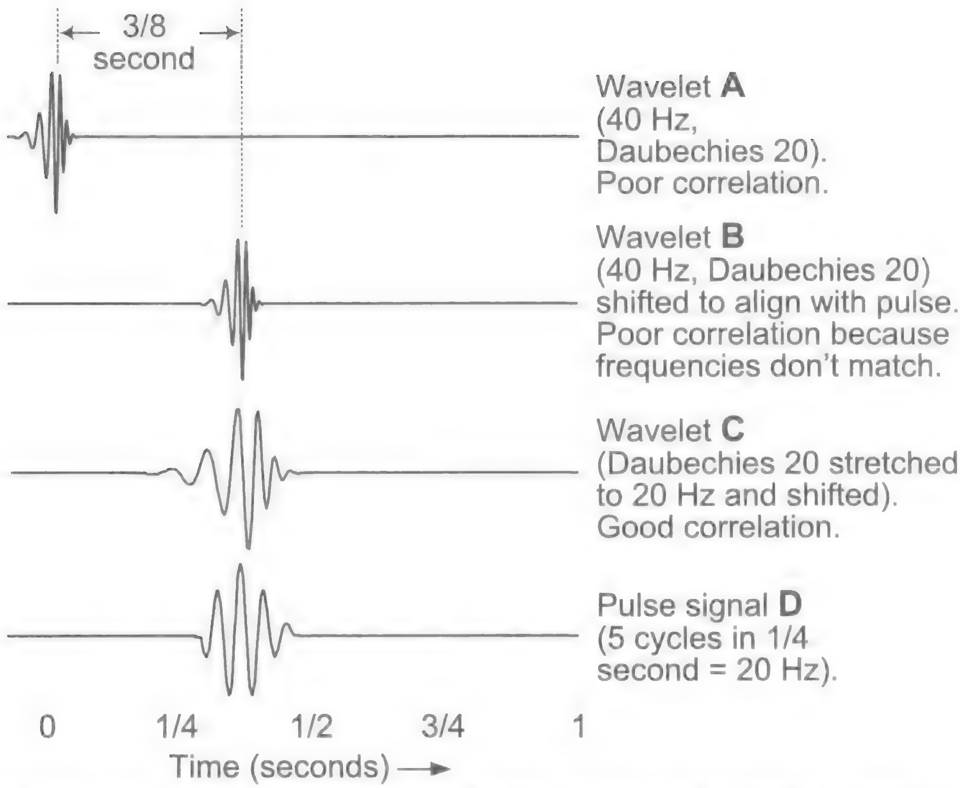


Figure 7-3 *CWT-type comparison of pulsed signal with several stretched and shifted wavelets.*

In the previous FFT discussion, we proceeded directly to stretching. In wavelet transforms, we shift the wavelet slightly to the right and perform another comparison with this new waveform to get another correlation value. We continue to shift until the Daubechies 20 wavelet is in the three-eighths-of-a-second time position shown in B. We get a little better comparison than A, but it's still very poor because B and D are different frequencies.

After we have shifted the wavelet all the way to the end of the 1-second time interval, we start over with a slightly stretched wavelet at the beginning and repeatedly shift to the right to obtain another full set of these correlation values. C shows the Daubechies 20 wavelet stretched to where the frequency is roughly the same as the pulse (D) and shifted to the right until the peaks and valleys line up fairly well. At this particular shifting and stretching, we should obtain a very good comparison and large correlation value. Further shifting to the right, however, even at this same stretching, will yield increasingly poor correlations.

In the continuous wavelet transform (CWT), we have one correlation value for every shift of every stretched wavelet. To show the correlation (comparison) results,

of all these stretches and shifts, we use a three-dimensional display with the stretching (roughly the inverse of frequency) as the vertical axis, the shifting in time as the horizontal axis, and brightness to indicate the strength of the correlation. Figure 7-4 shows a continuous wavelet transform display for the Figure 7-3 pulse signal (D). Note the strong correlation (bright areas) of the peaks and valleys of the pulse with the Daubechies 20 wavelet, the strongest (brightest) being where all the peaks and valleys best align.

Figure 7-4 shows that the best correlation occurs at the brightest points, between one-quarter and one-half of a second. This agrees with what we already know about the pulse (D). Figure 7-4 also tells us how much the wavelet had to be stretched (or *scaled*) and this indicates the approximate frequency of the pulse in our pulse signal. Thus, we know not only the frequency of the pulse, but also the time of its occurrence!

We run into this simultaneous time/frequency concept in everyday life. For example, a bar of sheet music may tell the pianist to play a C chord of three different frequencies at exactly the same time on the first beat of the measure.

For the simple Figure 7-3 example, we could have just looked at the pulse (D) to see its location and frequency. The next example is more representative of wavelets in the real world.

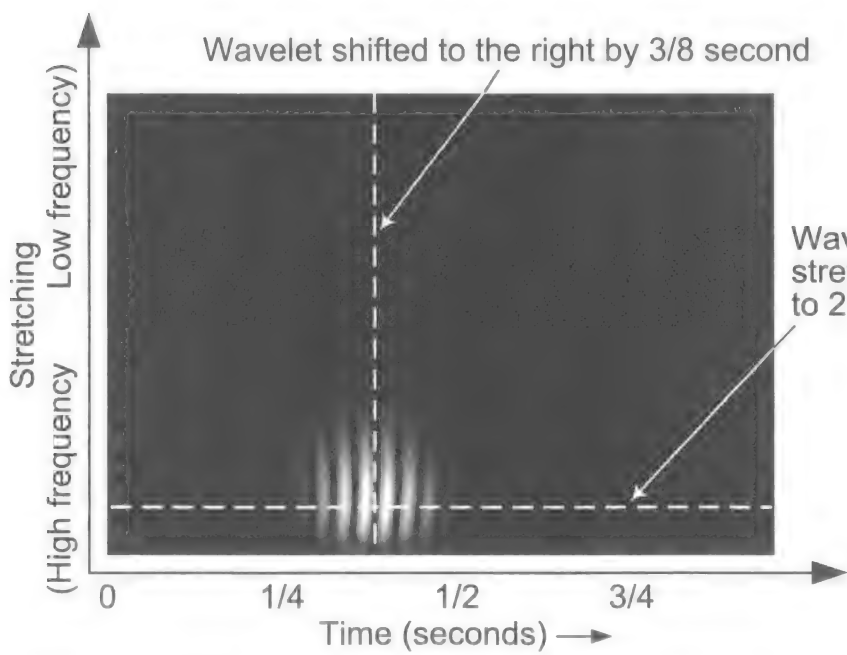


Figure 7-4 CWT display indicating the time and frequency of the pulse signal. (White bands show at what time the peaks and valleys of both the signal and the wavelet are aligned.)

Figure 7-5(a) shows a sine wave signal with a very small, very short-time discontinuity at Time = 180. The amplitude versus time plot of the signal does not show the tiny *event*. A standard fast Fourier transform (FFT) amplitude versus frequency plot would tell us what frequencies are present in the imperfect sine wave signal but would not indicate at what value of time those frequencies existed.

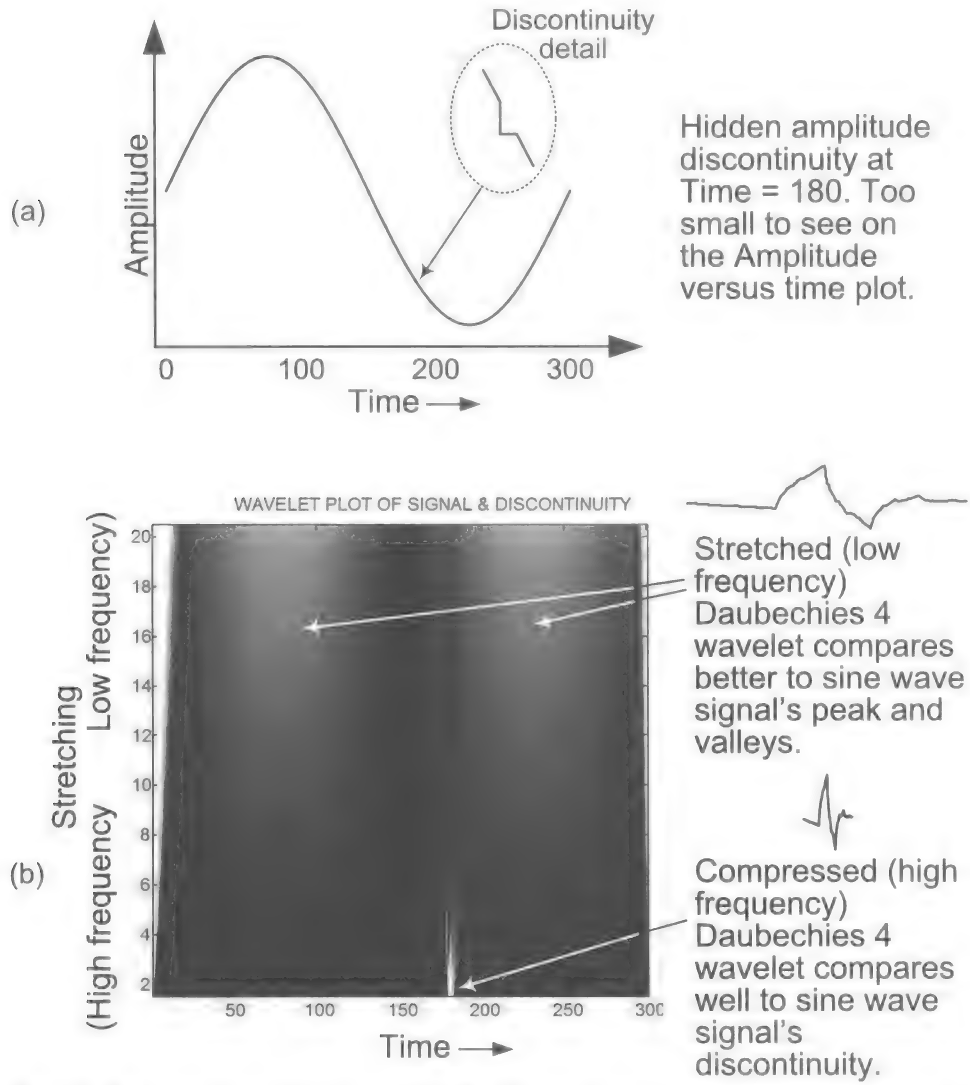


Figure 7-5 Detecting a small amplitude discontinuity: (a) sine wave signal containing a hidden discontinuity at Time = 180; (b) CWT of a sine wave signal with a hidden discontinuity.

With the wavelet display in Figure 7-5(b), however, at the bottom of that figure we can clearly see a vertical white band at Time = 180 at low scales when the wavelet has very little stretching, indicating a very high frequency. But more important to us, this tiny discontinuity has been precisely identified in time. The CWT display also *finds* the large oscillating wave at the higher scales where the wavelet has been stretched and compares well with the lower frequencies. For this short discontinuity, we used a short wavelet called a *Daubechies 4* wavelet for best comparison.

This is an example of why wavelets have been referred to as a *mathematical microscope* for their ability to *find* interesting events of various lengths and frequencies hidden in signal data.

By the Way

The continuous wavelet transform as computed on a digital computer is, of course, not really *continuous* in the analog signal sense we have discussed in this book. In wavelet jargon, it means that we compute so many more discrete sequence samples (all the possible integer factors of shifting and stretching) than in the short-sequence discrete wavelet transforms we are about to discuss, that it seems continuous by comparison.

You will encounter some very creative jargon in wavelets because established words have been given new definitions. Other examples, besides *continuous*, meaning more numerous, include: *decimation by 2* (having nothing to do with the *deci-* prefix); *dilation*, meaning to make either larger or smaller; *translation*, meaning shifting; *scaling*, meaning stretching; and family-friendly terms such as *mother*, *father*, and *daughter* wavelets! As Humpty Dumpty said in *Through the Looking Glass*, "When I use a word, it means just what I choose it to mean—neither more nor less."

Besides acting as a microscope to find hidden events in our signal data, wavelets can also separate the data into various frequency components, as does the fast Fourier transform (FFT). The FFT is used extensively to remove unwanted noise that is prevalent throughout the entire signal such as unwanted 60 Hz audio hum. Unlike the FFT, however, the wavelet transform allows us to remove frequency components at specific times within the signal data. This allows us a powerful capability to throw out the bad and keep the good part of the data in that frequency range.

The wavelet transforms we're about to discuss are called *discrete wavelet transforms* (DWT). They also have easily computed inverse discrete wavelet transforms (IDWT) that allow us to reconstruct the signal after we have identified and removed the unwanted noise or superfluous signal data in noise-removal or image-compression applications.

Undecimated or Redundant Discrete Wavelet Transforms (UDWT/RDWT)

One type of DWT is the *redundant discrete wavelet transform* (RDWT), often called the *undecimated discrete wavelet transform* (UDWT) for reasons we will soon see. With the RDWT, we first compare (correlate) the wavelet filter with itself. This produces a *highpass half-band filter* or *superfilter*. When we compare or correlate our signal with this superfilter, we extract the highest half of the frequencies. For a very simple denoising, we could just discard these high frequencies (for whatever time period we choose) and then reconstruct a denoised signal.

Multi-level RDWTs allow us to stretch the wavelet, similar to what we did in the CWT, except that it is done by factors of 2 (2 times as long, 4 times as long, etc.). This allows us stretched superfilters that can be half-band, quarter-band, eighth-band, and so forth.

Conventional (Decimated) Discrete Wavelet Transforms (DWT)

We stretched the wavelet in the CWT and the RDWT. In the conventional DWT, we shrink the signal instead and compare it to the unchanged wavelet. We do this by decimating by 2. Every other point in the signal is discarded. We have to deal with aliasing (not having enough samples left to represent the high-frequency components and thus producing a false signal). We must also be concerned with *shift invariance*. (Do we throw away the odd or the even values? It matters!)

If we are careful, we can deal with these concerns. One amazing capability of the filters in the conventional DWT is alias cancellation where the basic wavelet and three similar *filters* combine to allow us to reconstruct the original signal perfectly. The stringent requirements for the wavelets to be able to do this are part of why they often look so strange, as we shall see.

As with the RDWT, we can denoise our signal by discarding portions of the frequency spectrum obtained from a conventional DWT—as long as we are careful not to throw away vital parts of the alias cancellation capability. Correct and careful decimation also aids with compression of the signal.

There is a very common operation used in the field of signal processing called image compression, which refers to reducing the size, measured in bytes, of a graphics (picture) file without unacceptable degradation of the image quality. Reducing an image's file size enables us to store more images in a given amount of hard-disk memory as well as minimizing the time required to download or send images over the Internet. Modern JPEG image compression uses wavelets to produce the results demonstrated in Figure 7-6. An original image is shown in Figure 7-6(a), and a

(a) (b)

Figure 7-6 JPEG image compression using biorthogonal 9/7 wavelets: (a) before compression; (b) after compression.

biorthogonal 9/7 wavelet-compressed version of that image is given in Figure 7-6(b). The big deal is that the left image file requires 157 times the number of bytes as the wavelet-compressed image on the right! Can you see any difference between the two images?

By the Way

Thinking about JPEG image files, you may have also heard of MPEG video files. The file-name extensions ".jpg" and ".mpg" on your computer are their abbreviations. These are industry-standard electronic file compression methods. JPEG stands for Joint Photographic Experts Group and MPEG stands for Motion Picture Experts Group.

There are many types of wavelets. One type comes from mathematical equations while a second type is built from basic wavelet filters having as little as two points (two sample values). The Daubechies 4, Daubechies 20, and biorthogonal wavelets are examples of this second type. Figure 7-7 shows a 768-point approximation of a continuous Daubechies 4 wavelet with the four filter points (plus 2 zero-valued points) superimposed.

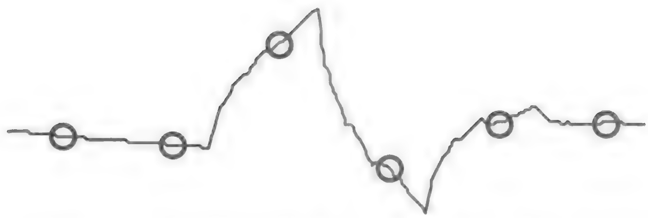


Figure 7-7 Daubechies 4 wavelet with four original filter points and two zero-valued end points.

Some wavelets have symmetry (valuable in human vision perception) such as the biorthogonal wavelet pairs. Shannon or *Sinc* wavelets can find events with specific frequencies. Haar wavelets (the shortest) are good for edge detection and reconstructing binary pulses. Coiflet wavelets are good for signal data with self-similarities (fractals) such as financial trends. Some of the wavelet families are shown in Figure 7-8.

You can even create your own wavelets, if needed. However, there is an embarrassment of riches (too much of a good thing) in the many wavelets that are already out there and ready to use. We have already seen that, with their ability to stretch and shift, wavelets are extremely adaptable. You can usually get by very nicely with choosing a less-than-perfect wavelet. The only *wrong* choice is to avoid wavelets due to an overabundance of wavelet choices.

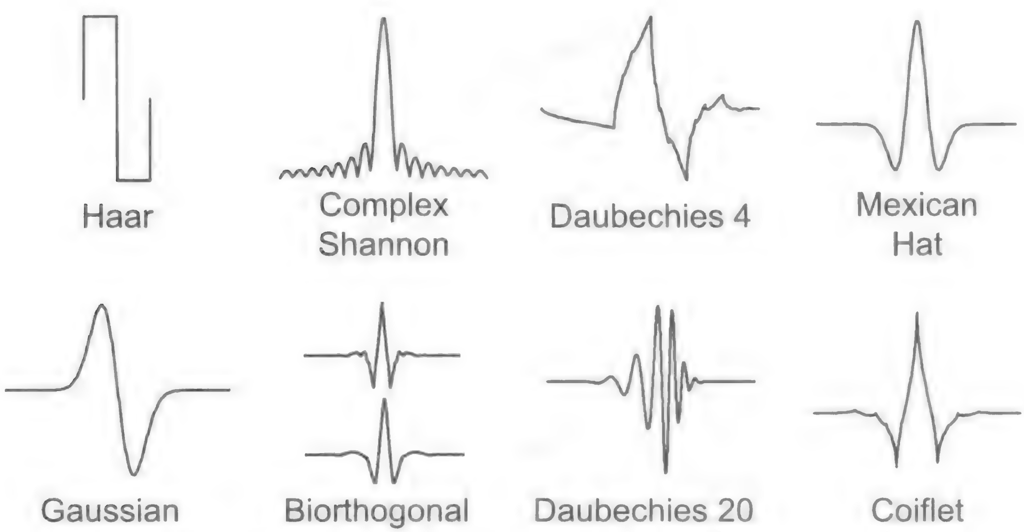


Figure 7-8 Examples of wavelet families.

The time spent in learning and correctly using wavelets for the fascinating real-world signals that have anomalies, or sudden changes in frequency at specific times, will be handsomely repaid. More information on wavelets can be found at http://www.ConceptualWavelets.com.

WHAT YOU SHOULD REMEMBER

The concepts you should remember from this chapter are:

- Wavelet transforms (wavelets) are mathematical processes allowing us to determine the spectrum of digital real-world signals having frequencies that change over time or have pulses, anomalies, or other *events* at certain specific times.
- Wavelets allow us to detect small amplitude discontinuities and other hidden events in digital signals.
- Wavelets are also used for image compression and are found in almost every modern digital camera and cell phone.
- There are many different types of wavelet transforms. They all stretch and shift and are very adaptable. However, one type may be better than another for a particular application (biorthogonal for images, Haar for short events, Daubechies 20 for chirp signals, Shannon for precise frequency determination, and so on).

8 Digital Filters

In previous chapters, we mentioned the idea of filtering analog and digital signals. Because filtering is a crucial operation in all voice communications, music, and video signal processing systems, let's learn more about filters and filtering.

The word **filter** means just what you might expect—a device that allows certain things to pass through and blocks other things. For example, the paper filter in your electric coffee maker allows water to pass through but blocks the coffee grounds. The air filter in your automobile allows air to pass through to your car's engine but blocks any dust particles.

Instead of water or air, let's talk about electronic **filters** that allow certain signal frequencies to pass and block other signal frequencies. In analog signal processing, an analog filter accepts an input analog voltage signal that contains some arbitrary amount of energy at various frequencies and only allows signal energy within a certain frequency range to pass through. In digital signal processing, a digital filter accepts an input digital signal sequence (a list of numbers) that contains some arbitrary amount of energy at various frequencies and only allows signal energy within a certain frequency range to pass through. For example, we may have a voice signal that is contaminated with high-frequency noise. Passing that noisy signal through a filter can eliminate the noise and yield a *clean* (uncontaminated) voice signal. Let's clarify this idea of filtering by looking at examples of both analog and digital filters.

ANALOG FILTERING

Analog filters are a collection of interconnected electronic hardware components mounted on a printed circuit board. These filters accept analog voltage signals as inputs and produce altered analog voltage signals as outputs.

As an example of analog filtering, let's say an engineer wants to build an analog sine wave generator whose frequency is 3 kHz (3,000 Hz). But due to hardware component imperfections, the engineer's generator output voltage is a badly distorted

analog signal as shown in Figure 8-1(a). Looking at the distorted sine wave's spectrum using an analog spectrum analyzer shows the desired 3 kHz sine wave signal is contaminated with undesired spectral components whose frequencies are 5 kHz and 7 kHz as presented in Figure 8-1(b).

One solution to the engineer's sine wave generation problem is to apply the distorted sine wave voltage to an analog lowpass filter that passes the desired 3 kHz

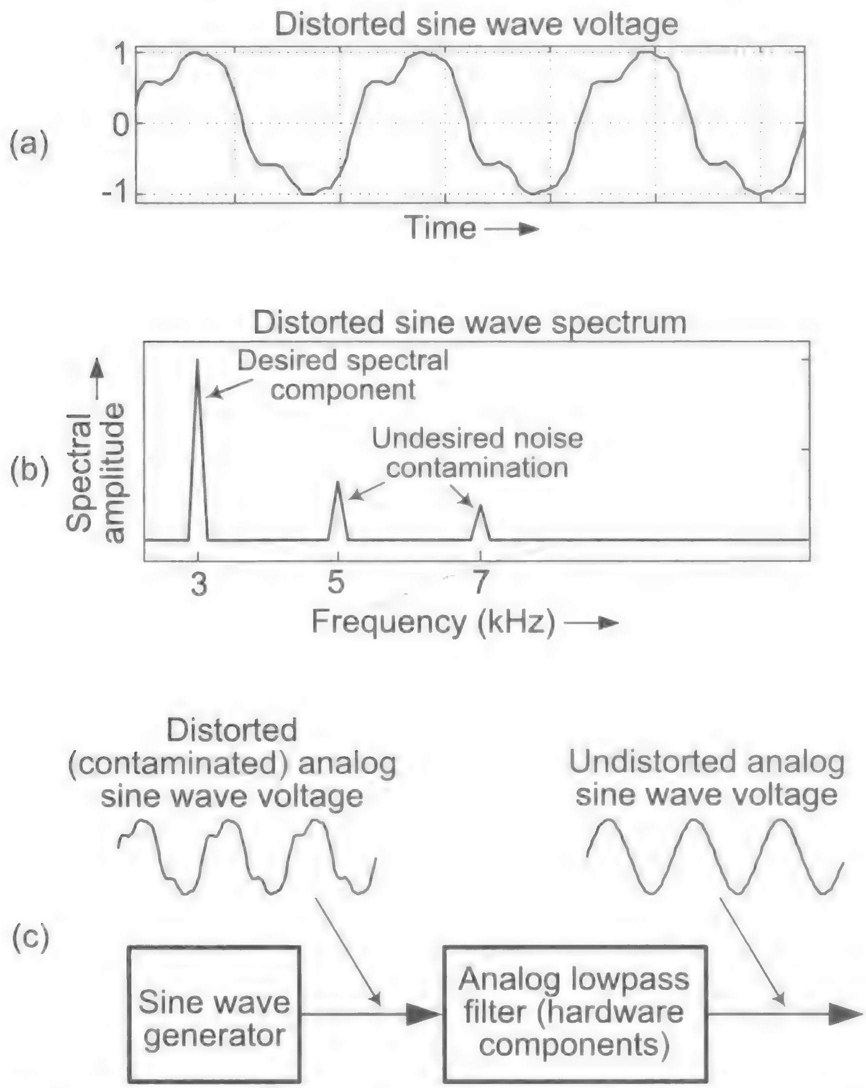


Figure 8-1 A distorted 3 kHz analog sine wave: (a) distorted sine wave voltage; (b) distorted sine wave spectrum; (c) lowpass filtering to produce an undistorted 3 kHz analog sine wave signal.

spectral energy and blocks the higher-frequency 5 kHz and 7 kHz spectral energy. This scenario is shown in Figure 8-1(c).

If you recall, we've already discussed one application of an analog lowpass filter. That was the analog lowpass anti-aliasing filter in Chapter 5's Figure 5-25(b) digital signal spectrum analysis example.

GENERIC FILTER TYPES

The filter in Figure 8-1(c) is called a **lowpass filter** because it passes low frequencies and blocks high frequencies. We show the characteristic of a generic lowpass filter as the frequency-domain curve in Figure 8-2(a). The **passband** of that curve is the frequency range of signals that will pass through the filter. The **stopband** of the curve is the frequency range of signals that will be blocked from passing through the filter. For our lowpass filter in Figure 8-1(c), 3 kHz is in the passband of the lowpass filter, and 5 kHz and 7 kHz are in the stopband of the filter.

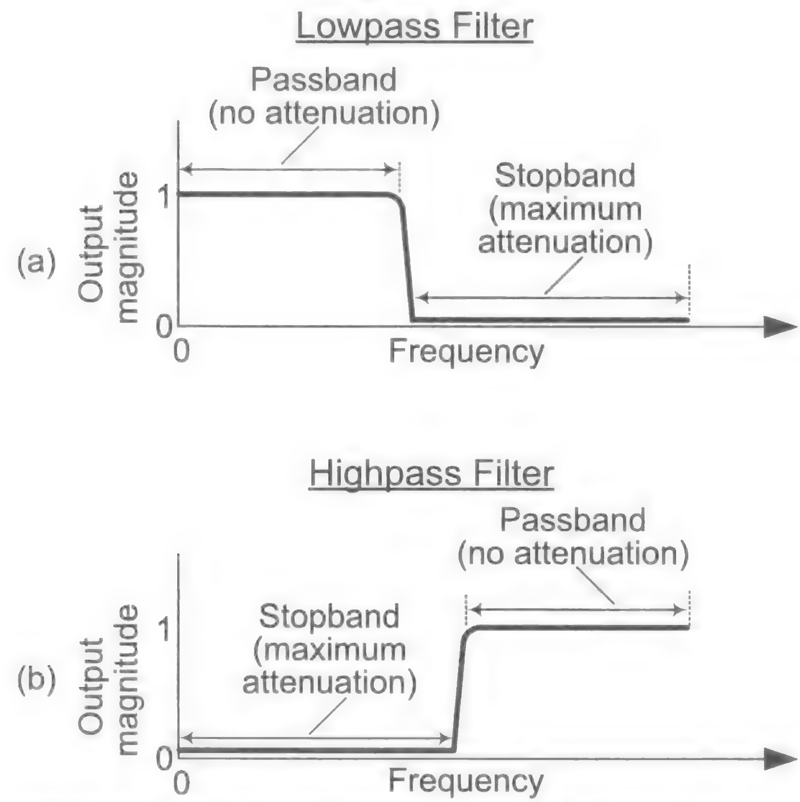


Figure 8-2 Generic filter frequency-domain behavior: (a) lowpass filter; (b) highpass filter.

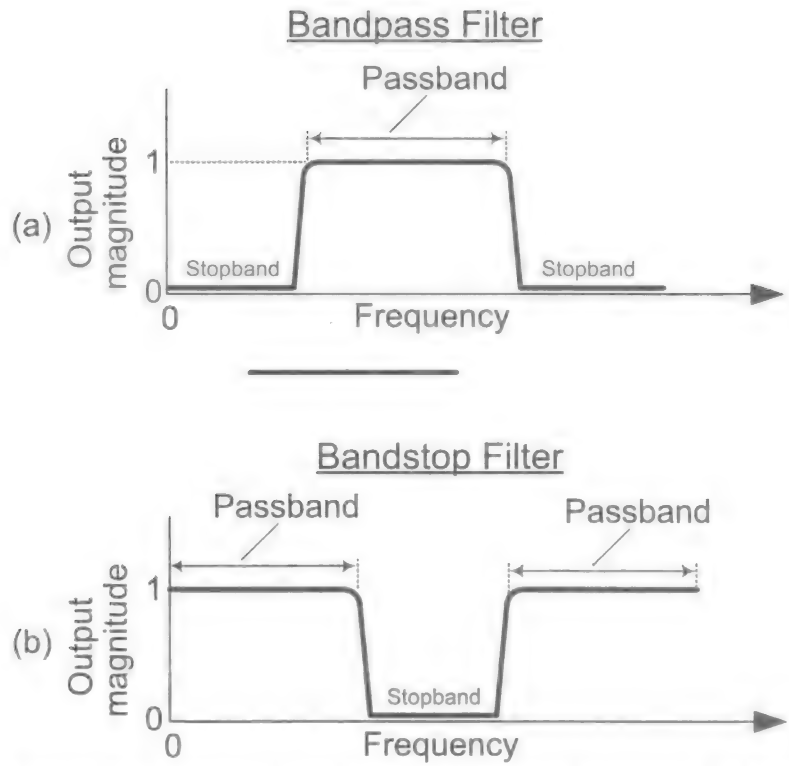


Figure 8-3 Generic filter frequency-domain behavior: (a) bandpass filter; (b) bandstop filter.

The frequency-domain behavior of a **highpass filter** is shown in Figure 8-2(b). This figure shows us that a highpass filter allows high-frequency signals to pass through, while it blocks low-frequency signals.

There are two other types of filters used in signal processing: **bandpass filters** and **bandstop filters**. The frequency-domain behavior of these filter types is shown in Figure 8-3. Bandpass filters have a passband between two stopbands, while stopband filters have a stopband between two passbands.

It's important to note that the four generic filter types described above, with their specific passband and stopband locations, can be implemented as either analog or digital filters.

DIGITAL FILTERING

Digital filters share all the behavioral characteristics of analog filters, with two exceptions: first, digital filters operate on digital signals (sequences of numbers); second,

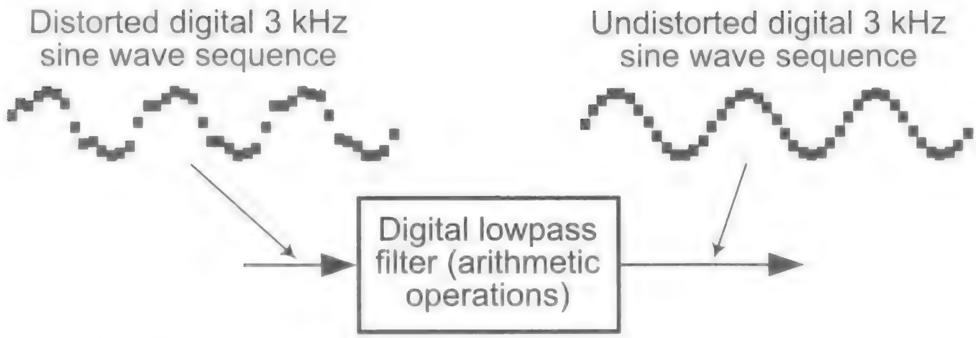


Figure 8-4 Digital lowpass filtering of a distorted 3 kHz digital sine wave signal.

digital filters are arithmetic operations rather than electronic hardware components. We illustrate these exceptions by showing a digital signal lowpass filtering process in Figure 8-4, where a noisy digital 3 kHz sine wave signal is filtered to produce a clean sine wave digital signal. That is, arithmetic operations are performed on the input digital signal sequence of numbers to generate a new digital signal sequence of numbers that we treat as the output of the digital filter. Let's now show the arithmetic details of a simple, real-world digital lowpass filtering example.

Doctors like to obtain their patients' accurate blood pressure readings. In addition to taking periodic blood pressure readings, your doctor might be looking for any trends such as a slow rise of blood pressure over, say, a year's time. A single blood pressure reading produces two numbers of interest to a doctor, the systolic and diastolic blood pressure values. However, for this digital filtering example we'll focus only on the first number, the systolic blood pressure.

Figure 8-5(a) shows a hypothetical sequence of 365 daily systolic blood pressure readings. That sequence is a digital signal and, for clarity, we plot it with straight lines rather than using dots.

It's pretty hard to spot any long-term blood pressure trends here because the reading values fluctuate so much, ranging from 101 to 166. The first four readings are 148, 107, 139, and 133. If we average those first four values, we obtain a result of 132. If we average the 2nd, 3rd, 4th, and 5th readings, we obtain a result of 124. When we average the 3rd, 4th, 5th, and 6th readings, we obtain a result of 129. Let's continue this process of averaging sets of four incremented successive readings for the whole year. This arithmetic process of averaging successive groups of four pressure values is called a **moving averager** and is one method of lowpass filtering signal data. Figure 8-5(b) shows the results of using this four-point moving averager.

We can see a little smoothing of the averaged values in Figure 8-5(b) where the value fluctuations are reduced. However, a doctor would still be hard pressed to spot any long-term trend in blood pressure readings. If we average a larger number of successive readings, we would achieve better smoothing. Averaging the 1st through the

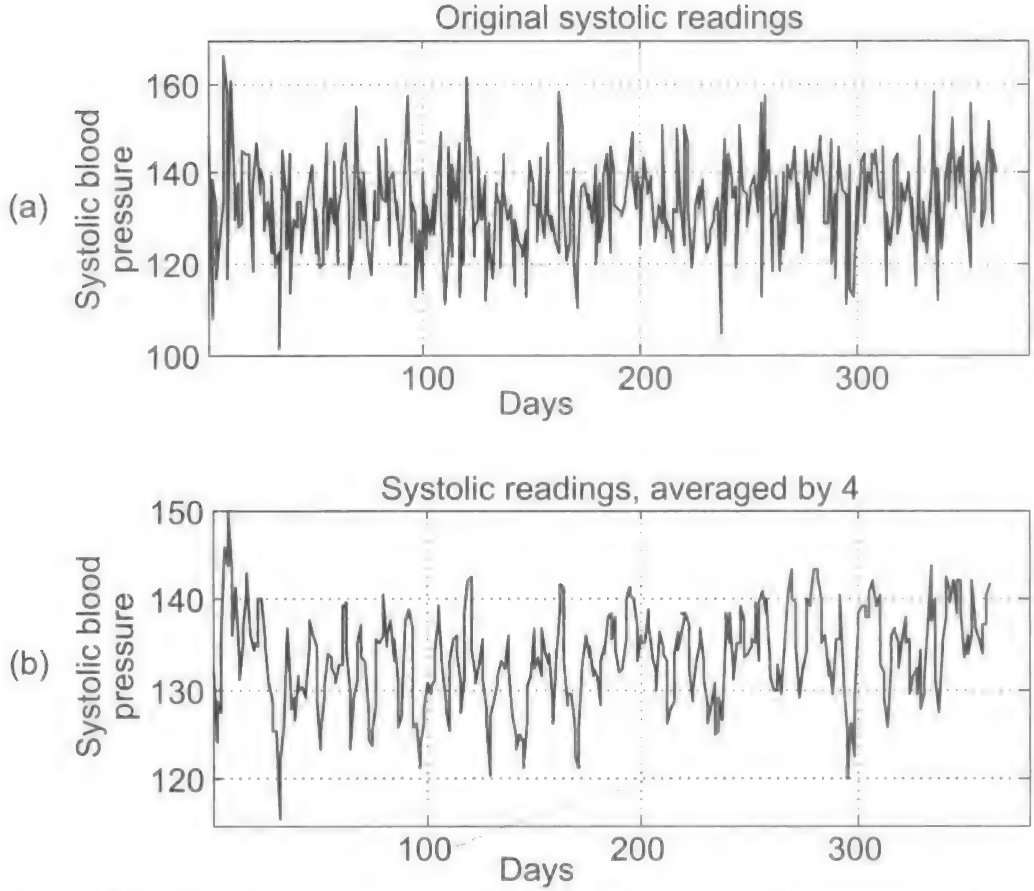


Figure 8-5 Blood pressure readings: (a) original readings; (b) averaged-by-4 results.

16th readings, the 2nd through the 17th readings, the 3rd through the 18th readings, and so on produces the sequence of values shown in Figure 8-6(a). There we see further data-value smoothing, in which case a doctor can begin to see an upward trend in blood pressure readings. What we are doing here is lowpass filtering the original blood pressure data sequence to reduce its high frequency value fluctuations.

Going further, we can average our original blood pressure readings in successive sets of 64 contiguous samples. That is, averaging the 1st through the 64th readings, the 2nd through the 65th readings, the 3rd through the 66th readings, and so on. Doing this results in the curve shown in Figure 8-6(b). The reading values are now significantly

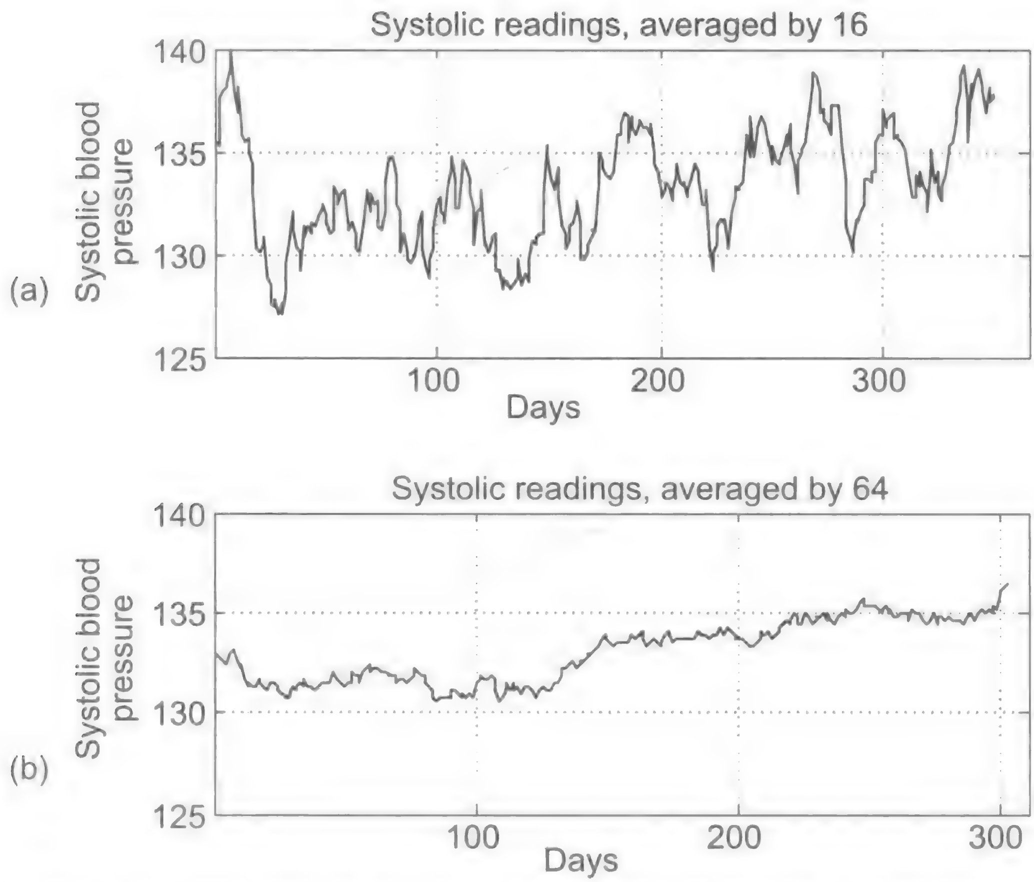


Figure 8-6 Blood pressure readings: (a) averaged-by-16 results; (b) averaged-by-64 results.

smoothed and a doctor can easily detect increasing blood pressure over the period of one year. Again, this long-term upward trend was not noticeable in the original blood pressure readings shown in Figure 8-5(a).

At the beginning of this section, we stated that digital filters are essentially arithmetic operations. And the blood pressure numerical averaging operations discussed above are forms of digital lowpass filtering. It's important to realize that there are many different arithmetic methods to implement digital filtering, some more complicated than others. The moving averager digital filter described above is the simplest of all digital filter implementations.

WHAT YOU SHOULD REMEMBER ....

The concepts you should remember from this chapter are:

- Filters are used to eliminate unwanted spectral frequencies in an input signal.
- An analog filter, a collection of interconnected electronic hardware components, operates on analog voltages. Analog-to-digital conversion is often preceded by analog lowpass filtering.
- A digital filter, an arithmetic process, operates on sequences of numerical data.
- There are many types of specialized filters, both analog and digital, in wide use in today's electronic systems.

9 Binary Numbers

In previous chapters, we've repeatedly stated that a digital signal generated by an analog-to-digital converter is merely sequences of numbers, as shown in Figure 9-1.

To be specific, the numbers comprising a digital signal are not the normal decimal numbers that we deal with in our daily lives. The numbers in a digital signal are called **binary numbers** and this chapter describes the characteristics, utility, and necessity of using binary numbers.

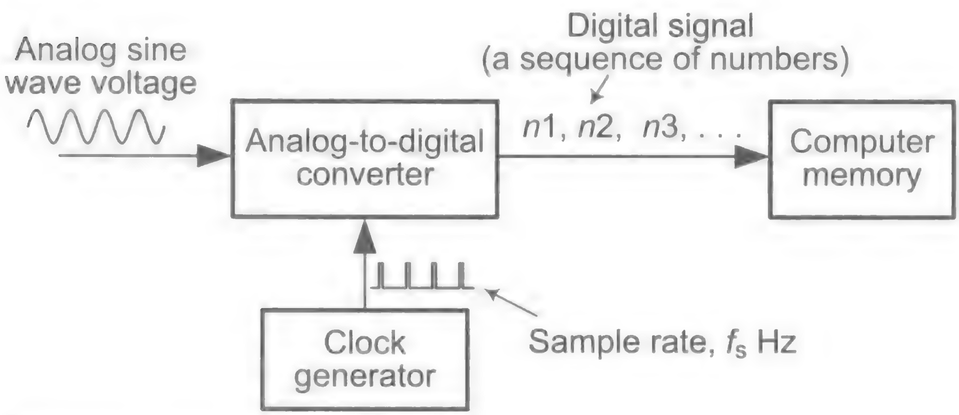


Figure 9-1 Digital signal generation.

NUMBER SYSTEMS

Understanding the nature of binary numbers is easy if we first recall what we learned about decimal numbers in grade school. Let's do that.

Decimal Numbers, a Base-10 Number System

In our decimal number system, we have 10 digits, 0 through 9, and we can represent any whole number using a series of decimal digits as shown in Figure 9-2. In that figure, we show the decimal number 1203, one thousand two hundred and three. Figures 9-2(a) and 9-2(b) remind us of the values of each decimal digit in our 4-digit number. The value of a single digit depends on its place, or position, within the 4-digit number. For example, the value of the "2" digit in our number is *not* 2. The value of the "2" digit in our number is 200. That is, the value of the "2" depends on its place within a multidigit decimal number. This scheme is called a *place-value* system of writing numbers.

We use the subscripted 10 in Figure 9-2(c), 1203_{10}, to remind us that our number 1203 is a decimal number. The numerical concepts in Figure 9-2 are so familiar to us that we use them effortlessly in our everyday life.

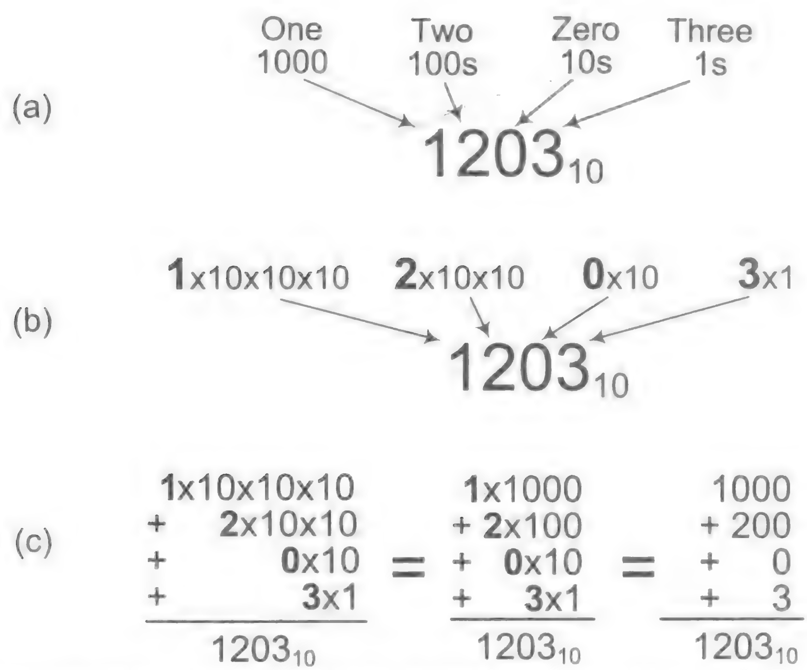


Figure 9-2 Digit values of the decimal number 1203_{10}.

Mathematicians refer to our familiar Figure 9-2 decimal number system as a **base-10 number system** because it has 10 different digits, 0 through 9.

A Base-4 Number System

To reinforce our understanding of the place-value method of writing numbers, in preparation for discussing binary numbers, let's consider a hypothetical base-4 number system that has only four digits, 0 through 3. Using the same place-value concepts in Figure 9-2, Figure 9-3 shows us how to interpret the number 1203_4, in a base-4 number system.

The value of the "2" digit in our 1203_4 number is 2 times 4 squared—that is, 2 times decimal 16. The value of the "1" digit in our 1203_4 number is 1 times 4 cubed, 1 times decimal 64. Figure 9-3(c) shows us that the decimal value of our base-4 1203_4 number is 99_{10}. If you understand the numerical notation conventions in Figures 9-2 and 9-3, pat yourself on the back. You're becoming well-versed in the mathematics of *place-value* number systems.

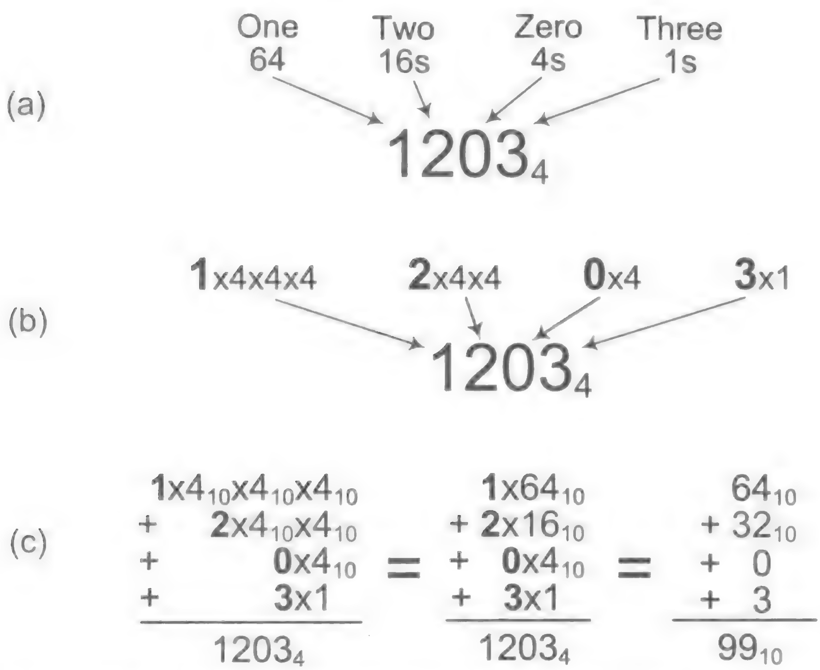


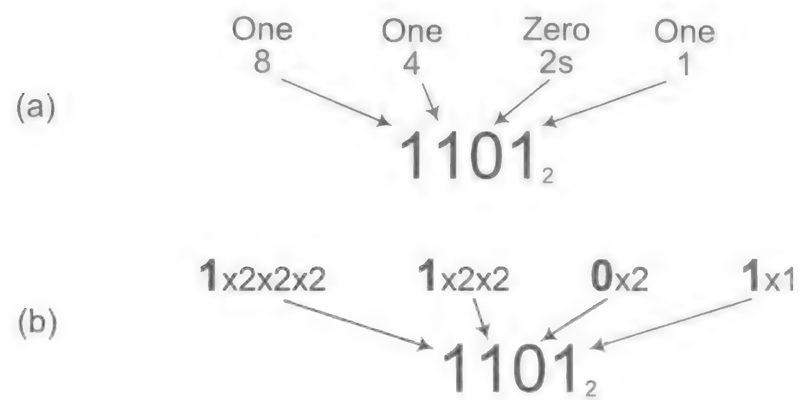
Figure 9-3 Digit values of the base-4 number 1203_4.

> **By the Way**
> The place-value system of representing numbers is very old—so old, in fact, that its origin is obscure. However, with its inherent positioning of the units digit as the right-most digit, this number system is so convenient and powerful that its importance has been compared to that of the first alphabets.

Binary Numbers, a Base-2 Number System

The **binary number** system—used by all computers, cell phones, compact discs, and hand calculators—is a base-2 system that has only two digits, 0 and 1. (The word *binary* comes from the Latin word *binarius*, which means consisting of two; for example, bicycle, bilingual, and binoculars.)

As a binary number example, Figure 9-4 presents the binary number 1101_2, where Figure 9-4(a) and Figure 9-4(b) show the decimal value of each digit in the binary number. Figure 9-4(c) shows us that our binary number 1101_2 has a decimal value of 13_{10}. We used the subscripted 2 in Figure 9-4 to make clear that our number 1101_2 is a binary number; a **base-2** number system. (Hardware and software engineers who routinely work with binary numbers don't bother using the 2 subscript.)



(c)

$$\begin{array}{r} 1\text{x}2_{10}\text{x}2_{10}\text{x}2_{10} \\ +\quad 1\text{x}2_{10}\text{x}2_{10} \\ +\quad 0\text{x}2_{10} \\ +\quad 1\text{x}1 \\ \hline 1101_2 \end{array} = \begin{array}{r} 1\text{x}8_{10} \\ +\ 1\text{x}4_{10} \\ +\ 0\text{x}2_{10} \\ +\ 1\text{x}1 \\ \hline 1101_2 \end{array} = \begin{array}{r} 8_{10} \\ +\ 4_{10} \\ +\ 0 \\ +\ 1 \\ \hline 13_{10} \end{array}$$

Figure 9-4 Digit values of the binary (base-2) number 1101_2.

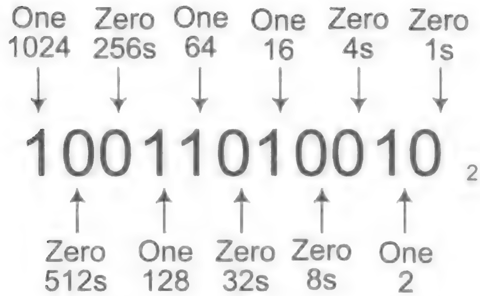
Table 9.1 First 16 Binary Numbers

Binary Number	Decimal Value
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Table 9.1 shows the first 16 binary numbers and their equivalent decimal values. The numbers in the table's left column are called 4-bit binary numbers because they each have 4 Binary digits. A **bit**, a single binary digit of either 0 or 1, is the smallest unit of information in a binary number.

As strange as they look, binary numbers are just as useful and practical as our decimal numbers. Any decimal number can be written in the binary number system. Figure 9-5 shows how to write the decimal number 1234 as a binary number. $1234_{10} = 10011010010_{2}$. All of the arithmetic operations that we perform on decimal numbers can also be performed on binary numbers, such as addition, subtraction, multiplication, and division.

What we're demonstrating here is that a sequence of binary bits (ones and zeros) can be used to represent a decimal number. And in that context, we call such a sequence of binary bits a binary number. However, as we discuss in the next section, we can use binary bits to represent other types of information.



$$10011010010_2 = 1024_{10} + 128_{10} + 64_{10} + 16_{10} + 2_{10} = 1234_{10}$$

Figure 9-5 Binary number equivalent of the decimal number 1234.

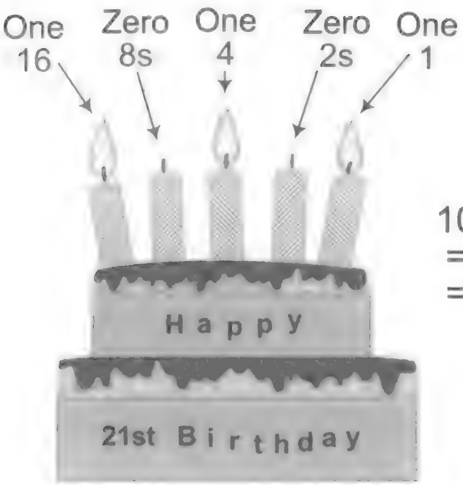
By the Way Now you're able to understand the computer geek joke, "There are only 10 kinds of people—those who understand binary numbers, and those who don't."

As it turns out, there are a number of different ways to represent decimal numbers with sequences of binary bits. For the interested reader, those representations, known as **binary number formats**, are presented in Appendix D.

Using Binary Numbers at Home

Let's use an example of a birthday party to show how the binary number system works. If you're celebrating someone's twenty-first birthday and only have five candles, there's no need to run to the store for more candles: binary numbers to the rescue. You merely arrange and light the candles as shown in Figure 9-6. A lit candle represents binary one bit and an unlit candle represents binary zero bit. This way, your five candles represent the decimal number 21. You can then explain binary numbers to your birthday party guests and proudly announce, "We often use the binary number system in this household."

By the Way Using the binary birthday candle technique means that all you'll ever need are seven candles. Seven lit candles, $1111111_2 = 127_{10}$, enable you to celebrate birthdays up to the age of 127 years. (This would cover even the ages of the authors of this book!)



$$10101_2 = 16_{10}+0+4_{10}+0+1 = 21_{10}$$

Figure 9-6 Celebrating a twenty-first birthday with five candles and binary numbers.

Table 9.2 Industry-Standard Binary Data Convention for Lowercase English Letters

English Letter	Binary Data	English Letter	Binary Data
a	01100001	n	01101110
b	01100010	o	01101111
c	01100011	p	01110000
d	01100100	q	01110001
e	01100101	r	01110010
f	01100110	s	01110011
g	01100111	t	01110100
h	01101000	u	01110101
i	01101001	v	01110110
j	01101010	w	01110111
k	01101011	x	01111000
l	01101100	y	01111001
m	01101101	z	01111010

BINARY DATA

A very common application of binary bits is to use them to represent letters. For example, Table 9.2 lists the industry-standard convention for representing lowercase English letters using 8 binary bits.

So if you use your computer's word-processing software to create the word *cat* and decide to print that word on your printer, your computer sends the three 8-bit binary data words 01100011, 01100001, and 01110100 by way of a cable to your printer. In turn, your printer is designed to interpret those three binary words as the English letters *c a t*, which it then prints.

By the Way

A sequence of 8 binary bits is called a **byte**. So if you have a digital photo whose file size is 50 kilobytes (50,000 bytes), that photo's file contains 400,000 binary bits. Likewise, an 8-gigabyte hard disk drive can store approximately 64 billion binary bits. And before you ask, yes, there is a **nibble**. It's half a byte, or 4 bits.

For another example of binary data, in a **high-definition television (HDTV)**, each tiny colored dot (called a **pixel**, short for *picture element*) on the television screen is controlled by three 8-bit binary words. Each 8-bit binary word represents the light intensity level of red, blue, and green colors that are combined to define the final color of a single pixel.

WHY USE BINARY NUMBERS? ....

At first glance, it's reasonable to ask why in the world would we use this apparently very limited binary number system to represent our decimal numbers. The answer lies in the practicality of building electric circuits that can represent the two binary digits of 1 and 0.

Digital Hardware Is Easy to Build

For all our power and sophistication, we use electric switches in digital hardware to represent a binary digit of 1 or 0. By electric switch we mean exactly the same function as the electric light switch on a wall. That is, if the switch is open (off), we interpret that to represent a binary 0 bit. And if the switch is closed (on), we interpret that to represent the binary bit of 1. (Even the power switches on many modern electrical appliances are labeled 1/0 instead of On/Off.) As such, to represent an arbitrary 8-bit binary word in hardware, we assemble a set of eight switches, each of which are either open or closed. However, the switches we use are not those familiar electrical light switches on a wall. In computer hardware, our switches are extremely small **transistors**, many millions of which can be fabricated) as a single **integrated circuit** (a **chip**), a miniaturized collection of interconnected (integrated) transistors

encapsulated in a small plastic or ceramic block. Digital engineers refer to these transistor switches as having two states, either opened or closed.

A reliable 0 to 9 electric circuit that has 10 states, so it can represent the 10 decimal digits of 0 to 9, could be built. But for a number of practical engineering reasons, the cost of such circuits would be far greater than binary circuits. The average person could never afford to buy a home computer, cell phone, or high-definition television built with decimal-like circuits.

By the Way

> Because a single transistor's On/Off state can represent a binary bit, a 0 or a 1, transistors are used for data storage in a computer's memory. A home computer's USB flash drive (thumb drive) memory device contains billions of transistors. In 2002, electronic industry analysts, who study such things, estimated there were more transistors produced that year than grains of rice, and the cost of one rice grain could buy hundreds of transistors. Astoundingly, in 2009 there were more than 250 transistors produced for each grain of rice, and the cost of one rice grain could buy 125,000 transistors.

Binary Data Is Resistant to Degradation

Another powerful advantage of using binary numbers to represent signals is that binary numbers (ones and zeros) can be reliably reproduced (copied). Let's explain that idea with an example.

Years ago, popular music was sold in the form of audio cassette tapes containing two spools on which a thin, magnetically coated plastic tape was wound. The analog music signal was represented by the intensity of the magnetism on the tape's magnetic coating. Now, if you recorded a second cassette tape from an original cassette tape, the second-generation music signal would be somewhat degraded in audio quality. That is, a noticeable low-level audio hiss could be heard in the background of the music on the second-generation cassette tape. And, if you recorded a third cassette tape from a second-generation cassette tape, the ever-increasing amount of audio hiss made the third-generation music signal generally unacceptable in audio quality.

Although cassette-player manufacturers produced dual-tape-drive products so people could make illegal bootleg copies of analog music cassette tapes, the practice never did become very popular.

Unfortunately for the commercial music industry, this situation drastically changed with the advent of compact discs (CDs) containing digital (binary) music signals. Because the digital signals on music CDs are nothing more than sequences of binary ones and zeros, *exact* duplication of such binary number sequences is simple and reliable. All we have to do is ensure that a one doesn't get copied as a zero, and vice versa. Because digital signals can easily be copied (reproduced) without error or

degradation, bootlegged music CDs have the identical high-fidelity audio quality of the original CDs from which they were copied.

This resistance to signal-quality degradation during recording, transmission, and reproduction is why today's music audio, telephone audio, Internet video, and television video signals are now implemented as digital signals rather than the old-fashioned analog signals of years past.

BINARY NUMBERS AND ANALOG-TO-DIGITAL CONVERTERS

As we stated at the beginning of this chapter, binary numbers have a special significance in the field of digital signal processing. That's because the numerical sample values of all digital signals obtained from analog-to-digital converters are in the form of binary numbers.

For example, in Chapter 4 we used Figure 4-14 and Figure 4-15 to show how the telephone company accepts an analog audio touch-tone keypad signal and converts it to a digital signal (a sequence of numbers). Telephone companies use analog-to-digital converters whose outputs are 8-bit binary numbers. Given what we now know about binary numbers, we can draw an 8-bit digital signal generation process as shown in Figure 9-7. In that figure, we see the individual 8 bits where each arrow exiting the analog-to-digital converter represents a single electrical wire. If there is a voltage on a wire, then that bit is a binary 1. If there is no voltage on a wire, then that bit is a binary 0. A given set of 8 binary bits represents a single number in the binary number system.

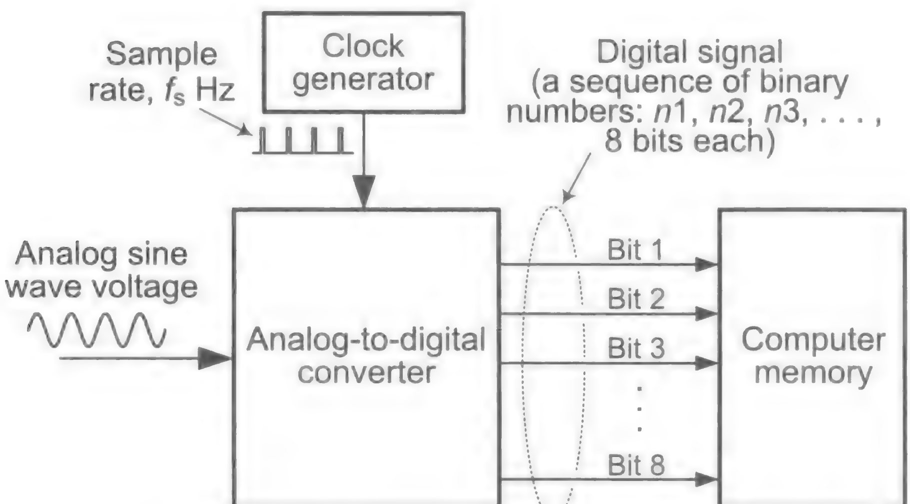


Figure 9-7 Digital signal generation showing individual 8-bit analog-to-digital converter output bits.

Commercial analog-to-digital converters are available with outputs that have anywhere from 6 bits to 24 bits. Given that, you might wonder how many bits an analog-to-digital converter should have for converting an analog signal to a digital signal. The answer lies in how accurately we want a digital signal to represent an analog signal.

For example, Figure 9-8(a) shows an analog voltage signal being converted to a 2-bit binary digital signal that is routed to a digital-to-analog converter producing a final analog voltage output. In that figure, we can see that using a 2-bit analog-to-digital converter produces a final output analog voltage that is a *very* crude version of the original input analog voltage signal. On the other hand, Figure 9-8(b) shows the same scenario using a 4-bit binary analog-to-digital converter. In that figure, we can see that using a 4-bit analog-to-digital converter produces a final output voltage that is somewhat similar to the original input analog voltage signal. Using 4-bit digital signals provided improved performance compared to using only 2 bits.

The choppy analog output analog signal in Figure 9-8(b) may be acceptable in some signal processing applications, but using only 4 bits for digital signals is

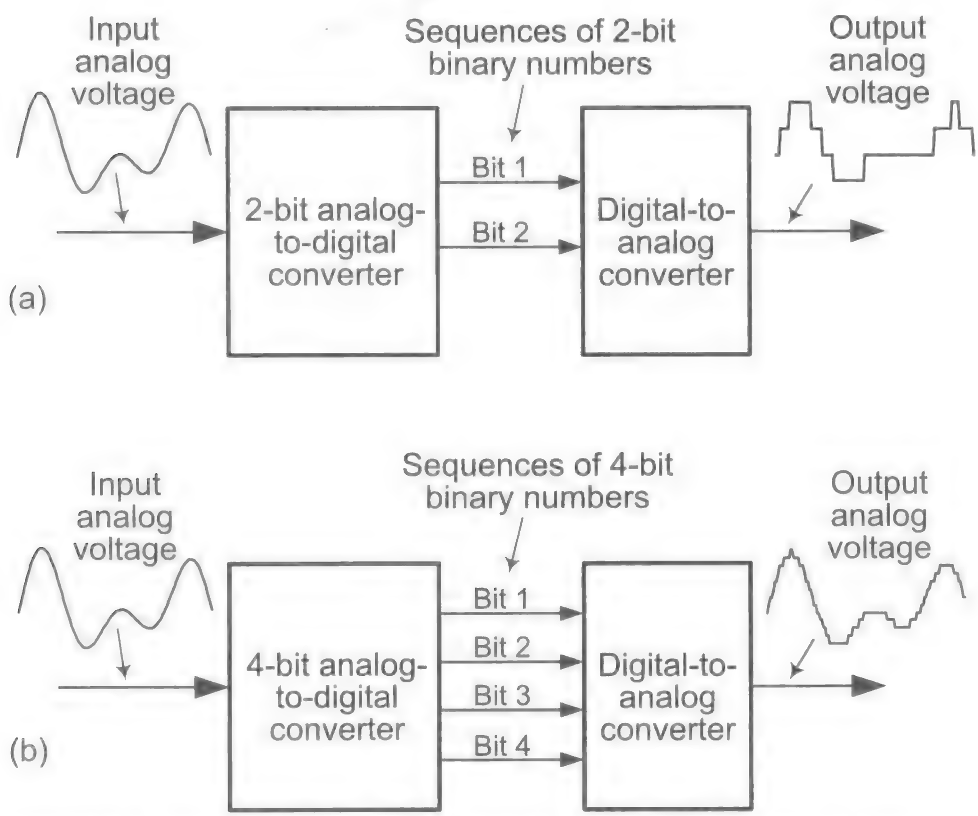


Figure 9-8 Performance in using 2- and 4-bit digital signals: (a) 2 bits; (b) 4 bits.

unacceptable in voice or music applications. If the analog output analog signal in Figure 9-8(b) was a human voice signal, you would hear the sound of a voice but you'd also hear an intolerable amount of background audio hiss (like the sound of loud static on an AM radio). Figure 9-9 shows the Figure 9-8 conversion scenario when 2-, 4-, 6-, and 8-bit binary numbers are used for the digital signal. In that figure, we see that 8-bit analog-to-digital conversion provides an output analog voltage that's *very* similar to the original input analog signal. That's why 8-bit analog-to-digital conversion is deemed acceptable for landline telephones.

So why don't we just use 24-bit analog-to-digital converters for all analog-to-digital conversion applications? The answer is, analog-to-digital converters that have more bits are more difficult to build and more expensive to buy. So if an application only requires an 8-bit analog-to-digital converter for acceptable operation, there's no need to spend extra money for unnecessary bits. For comparison purposes, Table 9.3 provides a short list of the number of binary bits used in various signal processing applications.

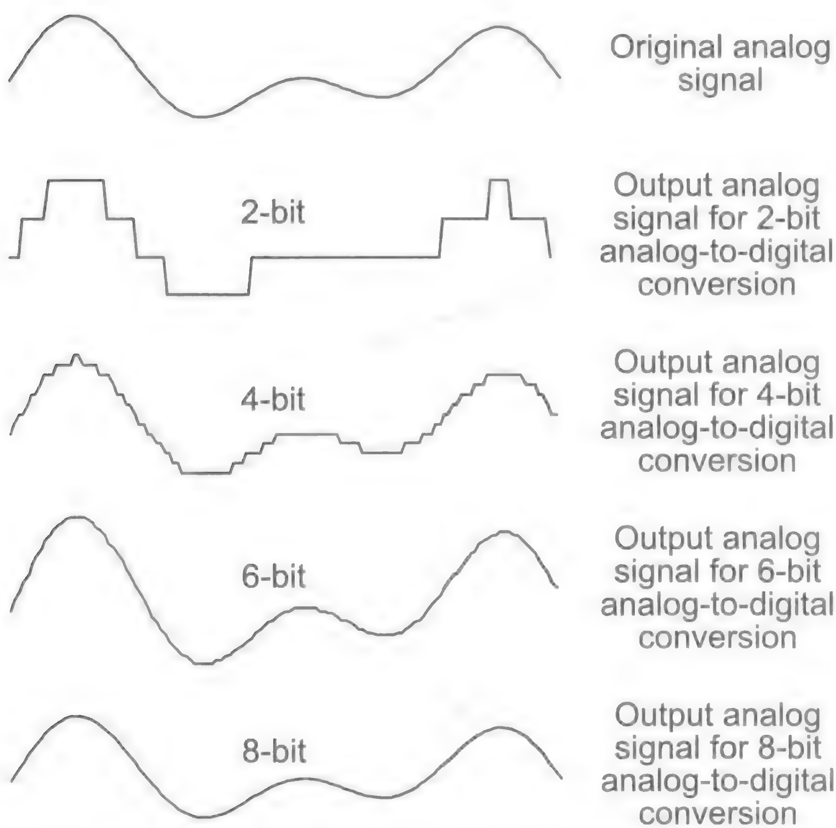


Figure 9-9 Output analog voltage signal quality for 2-, 4-, 6-, and 8-bit binary analog-to-digital conversion.

Table 9.3 Digital Signal Applications and Number of Bits Used to Represent an Analog Signal Value

Application	Number of Binary Bits
Telephones (landline)	8 bits
Audio recording function on home computers	16 bits
Music compact discs (CDs)	16 bits
Professional recording studio audio equipment	24 bits
Digital video discs (DVDs)	24 bits
Digital television	8 bits for each of the red, blue, and green colors of a single pixel
Ultrasound imaging	12 bits
Digital cameras	14 bits
Cellular phone base station	14 bits

WHAT YOU SHOULD REMEMBER ....

The concepts you should remember from this chapter are:

- Digital signals are sequences of numbers, and in digital hardware (computers, cell phones, digital cameras, high-definition televisions), those numbers are in the form of binary numbers.
- A binary number is a sequence of digits in which each digit can have only one of two possible values, zero and one. A single binary digit is called a bit.
- Binary numbers can be used to represent the decimal numbers that we're so familiar with.
- Binary numbers have the same *place-value* behavior as our everyday decimal numbers.
- A sequence of binary bits can also represent other information, such as letters of the alphabet.
- We use binary numbers in digital hardware to keep the hardware both reliable and inexpensive.
- The greater the number of bits used for digital signal samples, the more precise (more accurate) are those sample values (see Figure 9-9).

A Scientific Notation

Scientific notation is a convenient and precise way for engineers and scientists to represent very large and very small numbers. For example, the number 4,120,000 written in scientific notion is 4.12×10^6. The rules for scientific notation are given in Figure A-1.

In this example, we moved the implied decimal point from the right end of 4,120,000 six places to the left. By doing this, we divided the large number by one million so we have to multiply by a million, which is $10 \times 10 \times 10 \times 10 \times 10 \times 10$ or 10^6 to keep ourselves honest!

> **By the Way**
>
> In some cases, in displays that cannot show exponents (superscripts), the scientific notational number 4.12×10^6 is shown as 4.12e6. The "e" stands for exponent.

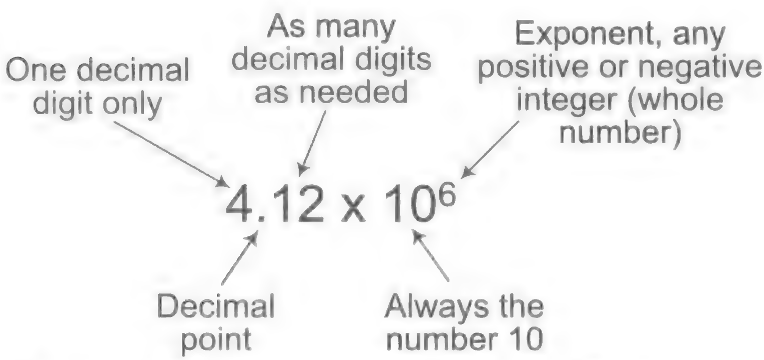


Figure A-1 The number 4,120,000 in scientific notation.

Table A.1 Scientific Notation Examples

Number	Scientific Notation
1,000,000	1×10^6
100,000	1×10^5
10,000	1×10^4
1,000	1×10^3
300	3×10^2
100	1×10^2
10	1×10^1 (not often used)
1	10^0 (not used)
0.1 (1/10)	1×10^{-1} (not often used)
0.03 (3/100)	3×10^{-2}
0.01 (1/100)	1×10^{-2}
0.001 (1/1000)	1×10^{-3}
0.0001 (1/1000)	1×10^{-4}

Table A.1 gives a few examples of numbers in scientific notation.

In scientific notation, it's easy to convert a number that's greater than 1 back to its long form as we show in Figure A-2. You write the number's decimal digits and move the decimal point to the right by the number of places specified by the original number's exponent, appending zeros where necessary.

Figure A-3 shows how to convert a number in scientific notation that's less than 1 back to its long form. There, we write the number's decimal digits and move the decimal point to the left by the number of places specified by the original number's exponent, inserting zeros where necessary.

$$4.12 \times 10^6 = 4.120000 = 4,120,000$$

1 2 3 4 5 6

Figure A-2 Converting a number greater than 1 from scientific notation to standard long notation.

$$5.76 \times 10^{-4} = 00005.76 = 0.000576$$

4 3 2 1

Figure A-3 Converting a number less than 1 from scientific notation to standard long notation.

Table A.2 Physical Constants in Scientific Notation Examples

Quantity	Long Notation	Scientific Notation
Speed of light	299,800,000 meters/second	2.998×10^{8} m/s
Transmission center frequency of a communications satellite	2,100,000,000 Hz	2.1×10^{9} Hz
Radius of an electron	0.00000000000000282 meters	2.82×10^{-15} meters
Diameter of the Milky Way galaxy	587,000,000,000,000,000 miles	5.87×10^{17} miles

Scientific notation may not look too convenient, but indeed it is. Table A.2 gives a few examples of real-world physical constants in scientific notation.

By the Way

All of the notation above is strictly English notation. Other languages have different numerical notation. For example, in the German language the numerical meanings of periods and commas are reversed from what they mean in English. The English number 3,425,978.64 is written as 3.425.978,64 in German. In our English scientific notation, that number is 3.42597864×10^{6}.

B Decibels

Signal processing engineers often need to measure the amplitude difference between two signals. For example, an engineer might need to compare the amplitude of a signal at the output of an amplifier to the amplitude of that signal at the input of the amplifier. Comparing the amplitude of those two signals describes the **gain** (the amount of amplification) of the amplifier. In practice, however, because various signals have such a wide range of amplitude values to be compared, engineers find it convenient to use **decibels** to simplify their numerical comparisons. This appendix describes how this is done.

The use of decibels is an expedient mathematical method for comparing the amplitude, or power, difference between two signals or between any two natural phenomena. That may sound a bit mysterious, but there's a good chance you're already familiar with use of decibels. We'll cover two examples of decibels that you are already acquainted with and then discuss decibels as they're used in signal processing.

Before we remind you of decibel values that you've encountered before, we simply must present one form of the equation for computing decibel values used to compare two numbers. That equation is:

$$\text{decibel comparison value} = 10 \cdot \log_{10}(P_1/P_2) \text{ dB} \tag{B-1}$$

We use the letters dB to represent the word *decibels*, much like we use the letters mph to represent the words *miles per hour*.

Don't be troubled by equation (B-1). (We won't ask you to compute any decibel values; we'll do any of the necessary calculations for you.) Stated in words rather than numbers, equation (B-1) is "the decibel value of number P_1 compared to number P_2 is 10 times the base-10 logarithm of the fraction P_1 divided by P_2." OK, with that said, let's look at a use of decibels that you've encountered in your daily life.

DECIBELS USED TO DESCRIBE SOUND POWER VALUES

Acoustic engineers, and some health officials, are concerned with measuring the power of audio sounds to which people may be exposed. And because the range of power (the loudness) of various sounds we might be subjected to in our daily lives is so large, decibels are used by technical people to categorize the power of those sounds. It's quite possible that, at one time or another, you have encountered the sound power values listed in the third column of Table B.1.

Acoustic engineers use their audio test equipment to measure the sound power value (in watts) in a particular environment, and declare that power value as number P_1 in equation (B-1). By mutual agreement in the audio engineering field, the value of 0.000000000001 watts (10^{-12} W) is defined as the number P_2. (See Appendix A for an explanation of that 10^{-12} notation.) Then, numbers P_1 and P_2 are inserted in equation (B-1) to compute a sound value measured in dB, such as the values in the third column of Table B.1. Again, decibels are used because it's easier to discuss, interpret,

Table B.1 *Acoustic Decibel Values of Common Sounds*

Sound Source	Sound Power in Watts (W)	Sound Power in dB
Turbojet engine	10,000 W	160 dB
Emergency vehicle siren	1,000 W	150 dB
Heavy truck engine or rock concert loudspeakers	100 W	140 dB
Machine gun, large pipe organ	10 W	130 dB
Jackhammer, small aircraft engine	1 W	120 dB
Trumpet	0.3 W	115 dB
Chain saw	0.1 W	110 dB
Helicopter	0.01 W	100 dB
Lawn mower, kitchen blender	0.001 W	90 dB
Dishwasher, alarm clock	0.0001 W	80 dB
Vacuum cleaner	0.00001 W	70 dB
Normal conversation	0.000001 W	60 dB
Average home, quiet office	0.0000001 W	50 dB
Refrigerator	0.00000001 W	40 dB
Quit library, whisper	0.000000001 W	30 dB

and document the simpler dB values than the clumsy, wide-ranging center-column wattage numbers in Table B.1.

The main thing to notice in Table B.1 is that a difference of 10 dB, in the rightmost column, means a sound power difference factor of 10. This means that a difference of 20 dB is a sound power difference by a factor of 100. So standing near a running lawn mower (90 dB) is 100 times louder than standing near a running vacuum cleaner (70 dB).

By the Way

In the early twentieth century, signal processing people compared the power of two signals using the equation

$$\log_{10}(P_1/P_2) \text{ bels}$$

where the unit *bel* was named in honor of the American inventor of the telephone, Alexander Graham Bell. The unit of bel was soon found to be inconveniently large. For example, it was discovered that the human ear could detect audio power value differences of one-tenth of a bel. Measured power value differences smaller than one bel were so common that it led to the use of the decibel (bel/10), effectively making the unit of bel obsolete.

Let's look at another common use of decibels that may be familiar to you.

DECIBELS USED TO MEASURE EARTHQUAKES.......

When you hear in the news of an earthquake that has happened somewhere on our planet, you're likely to hear the total energy of that earthquake specified by a value on what's called the Richter scale. Seismologists also use a logarithmic equation to compute Richter scale values, similar to our sound power values in dB. That Richter scale equation is

$$\text{Richter scale value} = \log_{10}(E_1/E_2) \tag{B-2}$$

We need not worry about the meaning of the energy values E_1 and E_2 in equation (B-2). We presented equation (B-2) merely to show that the mathematical operation of logarithms is used to categorize earthquake energies in the same way logarithms were used to compute sound power values in equation (B-1) and Table B.1.

Table B.2 presents a list of common Richter scale values for earthquakes. The center column of the table gives an estimate of an earthquake's total energy in terms of the equivalent explosive energy of tons of TNT (similar to dynamite).

The main thing to notice in Table B.2 is that a Richter value difference of 1, in the leftmost column, means an energy difference factor of 10. This means that a Richter scale difference of 2 is an energy difference by a factor of 100. So an 8.0 earthquake is 100 times as destructive as a 6.0 earthquake.

Table B.2 Richter Scale Values and Typical Effects

Magnitude Value	Equivalent TNT Seismic Energy	Typical Effects
Less than 2.0	Less than 1,000 pounds	Micro-earthquakes, recorded by seismographs but imperceptible by people.
2.0–2.9	1–29 tons	Rarely felt by people. No building damage.
3.0–3.9	32 to 900 tons	Often felt by people. No building damage.
4.0–4.9	1,000–29,000 tons	Noticeable shaking felt by people indoors. No building damage. Slightly felt outdoors.
5.0–5.9	32,000–900,000 tons	Felt by all people. Can damage poorly constructed buildings. Usually no damage to all other buildings.
6.0–6.9	1,000,000–29,000,000 tons	Strongly felt by all people. Some objects fall off indoor shelves. Some damage to moderately well-constructed buildings. Poorly constructed buildings suffer moderate to severe damage.
7.0–7.9	32,000,000,000–900,000,000 tons	Drastically felt by all people over a wide area (difficult to walk). All objects fall off indoor shelves. Damage suffered by most buildings near the quake's epicenter.
8.0–8.9	1,000,000,000–29,000,000,000 tons	Profoundly felt by all people over an enormous area, hundreds of square miles. Major damage suffered by most buildings, some totally destroyed.
9.0 and greater	Minimum of 32,000,000,000 tons	Catastrophic. Near or at total destruction; severe damage or collapse to all buildings. Damage and shaking extends to distant locations. Ground topography changed forever.

Similar to Table B.1's decibel (dB) sound power level values, it's easier to discuss, interpret, and document Table B.2's Richter scale values than the clumsy wide-ranging center column TNT energy values.

OK, now that we're familiar with converting wide-ranging numbers to simpler smaller-ranging numbers, let's see how signal processing engineers perform that same conversion.

DECIBELS USED TO DESCRIBE SIGNAL AMPLITUDES

As we stated at the beginning of this appendix, signal processing engineers use decibel values to conveniently compare the voltage amplitude levels of two signals. Similar to equations (B-1) and (B-2), signal processing folk use the equation

$$\text{signal decibel comparison value} = 20 \cdot \log_{10}(A_1/A_2)\ \text{dB}. \qquad \text{(B-3)}$$

to compute a decibel (dB) value that represents the relative amplitude difference between two signals. Value A_1 is the voltage amplitude of one signal and value A_2 is the voltage amplitude of the other signal that we're comparing to signal A_1. Before we look at an example of using decibels in signal processing, we present Table B.3 showing the relationship between the ratio of two signal amplitudes and equivalent decibel values. Spend a few moments reviewing Table B.3. Notice that when A_1 is less than A_2, ratio A_1/A_2 is less than 1, and the dB value is a negative number.

Table B.3 Relative Decibel Signal Levels

Amplitude Ratio (A_1/A_2)	Relative dB	Amplitude Differences
1/1000	−60	A_1 is one-thousandth of A_2
1/100	−40	A_1 is one-hundredth of A_2
1/10	−20	A_1 is one-tenth of A_2
1/4	−12	A_1 is one-fourth of A_2
1/2	−6	A_1 is one-half of A_2
1	0	A_1 is equal to A_2
2	6	A_1 is twice A_2
4	12	A_1 is four times A_2
10	20	A_1 is ten times A_2
100	40	A_1 is one hundred times A_2
1000	60	A_1 is one thousand times A_2

For example, here's how we interpret Table B.3: if the voltage amplitude of signal A_1 is one-tenth of the voltage amplitude of signal A_2, we say "the amplitude of A_1 is minus 20 dB relative to signal A_2." Signal processing engineers often speak in the language of dB.

As with the previous tables in this appendix, it's easier to discuss, interpret, and document Table B.3's center column dB values than the clumsy, wide-ranging amplitude ratios in the left column.

OK, let's now look at a simple example of using decibels in signal processing that illustrates the dB values in Table B.3. Figure B-1(a) shows a composite analog signal. We will call that signal by the name S_{in}. Signal S_{in} is the sum of a 2 kHz sinusoidal wave whose positive peak amplitude is 1, and a 3 kHz sinusoidal wave whose positive peak amplitude is 0.8. For illustrative purposes, we show the individual 2 kHz and 3 kHz sinusoidal waves in Figure B-1(b). Again, the Figure B-1(a) S_{in} signal is the sum of the solid- and dashed-line waves in Figure B-1(b). The spectrum of signal S_{in} is given in Figure B-1(c).

Let's assume we decide to remove (filter out) the 3 kHz component from the composite S_{in} signal. We can do that by passing signal S_{in} through the lowpass filter shown in Figure B-2(a). The lowpass filter, whose cutoff frequency is 2,500 Hz, operates as follows: any filter input-signal spectral energy with a frequency less than 2,500 Hz will pass through the filter and show up at the filter's output with no loss in amplitude. In addition, any filter input-signal spectral energy with a frequency greater than 2,500 Hz will pass through the filter and show up at the filter's output but with a significant loss in amplitude. That behavior is shown in Figure B-2(b).

Passing the Figure B-1(a) S_{in} signal through the lowpass filter produces an S_{out} output signal shown in Figure B-2(a), which looks like the 2 kHz solid-line curve in Figure B-1(b). It appears that the lowpass filter has eliminated the 3 kHz component. But how well did our filter really work? Let's say we apply the S_{out} signal to the input of a spectrum analyzer and obtain the display given in Figure B-3(b). There, we see that S_{out} still contains a *very* small amount of the 3 kHz sine wave. Using decibels, we can specify exactly to what degree our lowpass filter attenuated the undesired 3 kHz sine wave. Here's how.

From Figure B-1(c), the amplitude of the 3 kHz component of signal S_{in} at the filter's input is 0.8. We assign that amplitude value to the variable A_2 in equation (B-3). From Figure B-3(b), the amplitude of the undesired 3 kHz component of signal S_{out} at the filter's output is 0.008, so we assign that amplitude value to the variable A_1. The ratio A_1/A_2 now becomes $A_1/A_2 = 0.008/0.8 = 0.01 = 1/100$. From Table B.3, an A_1/A_2 ratio of 1/100 is equivalent to -40 decibels (dB). So there you have it. We can now describe the filter's performance in either of these two equivalent ways:

- The lowpass filter's stopband gain is -40 decibels (-40 dB).
- The lowpass filter attenuated signal S_{in}'s undesired 3 kHz component by 40 dB.

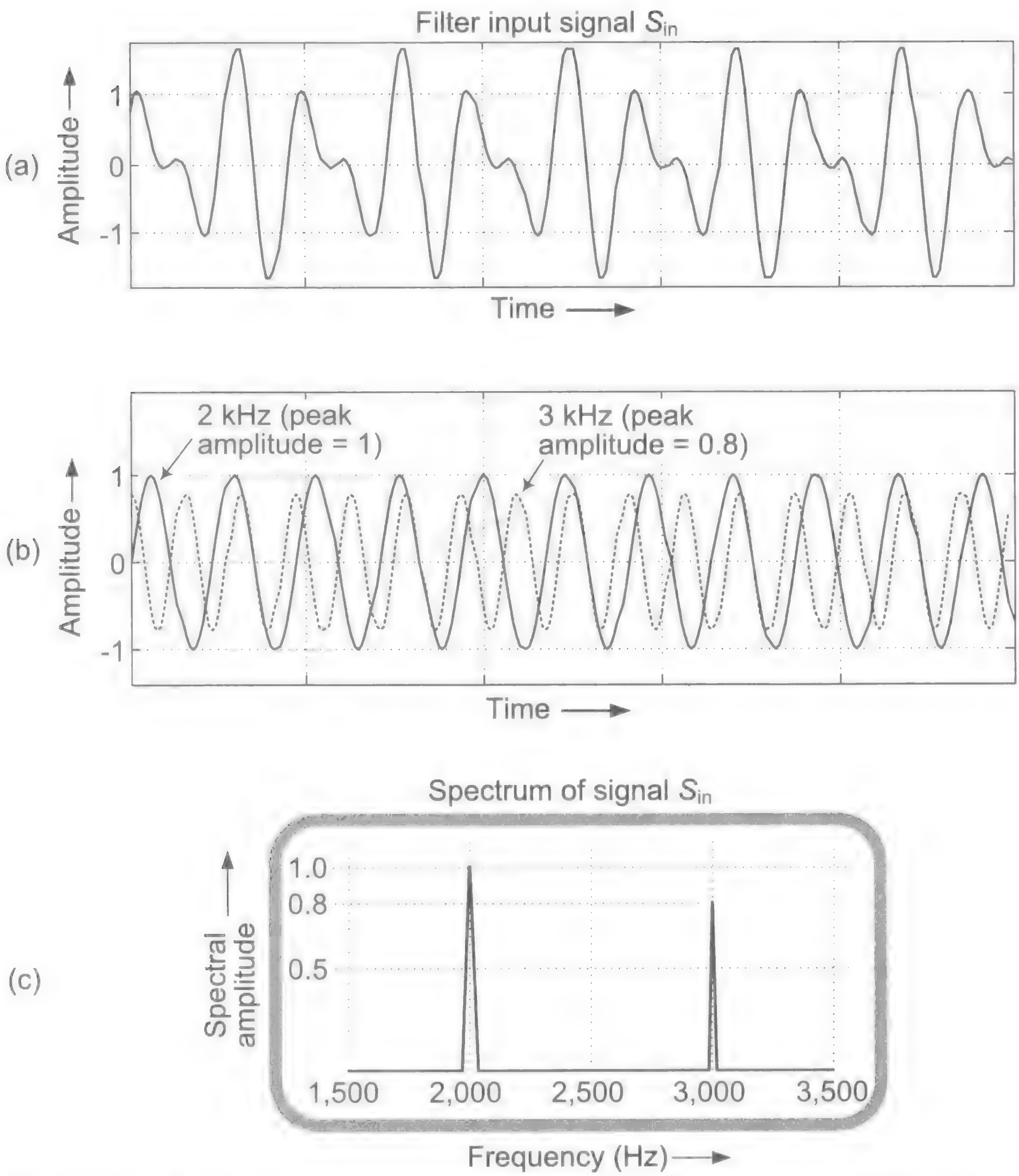


Figure B-1 Analog signal: (a) composite time signal S_{in}; (b) individual 2 kHz and 3 kHz time signals; (c) individual 2 kHz and 3 kHz spectral components.

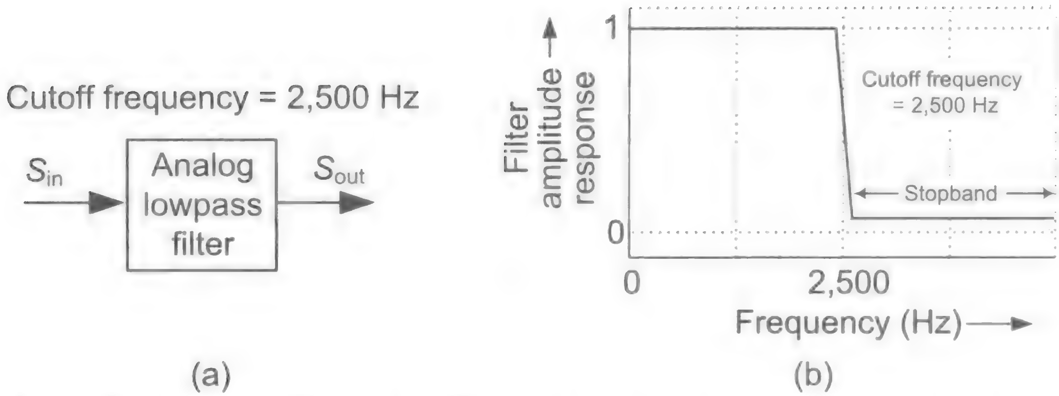


Figure B-2 Lowpass filtering: (a) filter input and output signals, lowpass filter cut-off frequency.

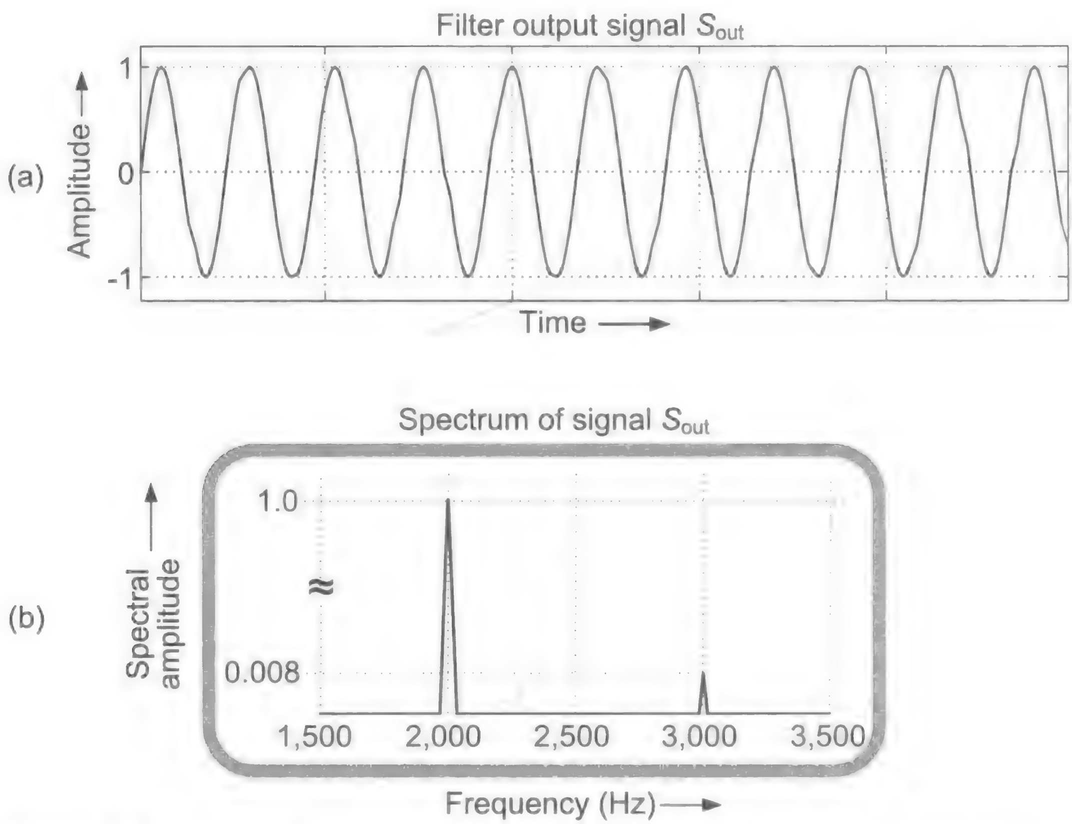


Figure B-3 Lowpass filtering:(a) individual 2 kHz and 3 kHz output spectral components; (b) output time signal S_{out}.

By the Way

On the off chance that you ever need to compute the relative dB difference between the amplitudes of two signals using equation (B-3), you don't necessarily need a scientific hand calculator. You can compute dB values using Microsoft Excel spreadsheet software. For example, to compute the decibel level of an amplitude of 0.008 divided by an amplitude of 0.8, in a cell of an Excel spreadsheet merely enter the following:

```
=20*LOG10(0.008/0.8)
```

Then hit your keyboard's Enter key. The number −40 will appear in that cell, meaning a decibel value of −40 dB.

DECIBELS USED TO DESCRIBE FILTERS...................

Decibel values are very often used to describe the performance of both analog and digital filters. Using equation (B-3), signal processing engineers plot a curve of a filter's frequency-domain behavior where the vertical axis is measured in decibels. An example of this, for a bandpass filter, is shown in Figure B-4.

Recall the lowpass filter in Figure B-2 that only allowed signals whose frequencies were greater than 2,500 Hz to pass through the filter. Figure B-4 is a bandpass filter that allows signals whose frequencies are within a frequency band to pass through the filter.

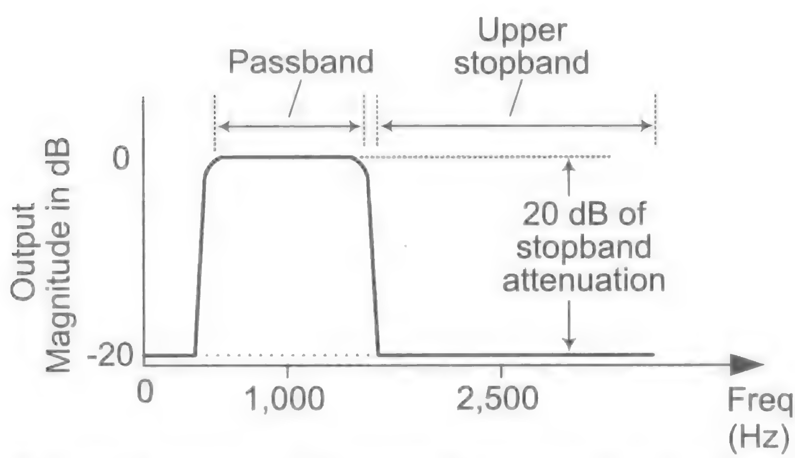


Figure B-4 Describing a bandpass filter's frequency response using decibels.

We interpret Figure B-4 as follows: Let's say we apply a 1,000 Hz sine wave, whose amplitude is A_2, to the input of the bandpass filter. According to Figure B-4, the A_1 amplitude of the 1,000 Hz filter output sine wave is zero decibels (dB) relative to the A_2 input amplitude. From Table B.3, we see that zero dB means that the ratio A_1/A_2 is equal to 1. So the 1,000 Hz output sine wave's amplitude is equal to the 1,000 Hz input sine wave's amplitude. The input 1,000 Hz sine wave arrived at the filter output with no reduction (no attenuation) in amplitude.

On the other hand, assume we now apply a 2,500 Hz sine wave, whose amplitude is A_2, to the input of the bandpass filter. According to Figure B-4, the A_1 amplitude of the 2,500 Hz filter output sine wave is –20 decibels (–20 dB) relative to the A_2 input amplitude. From Table B.3 we see that –20 dB means that the ratio A_1/A_2 is equal to 1/10. So the 2,500 Hz output sine wave's amplitude is equal to the 2,500 Hz input sine wave's amplitude divided by 10 (attenuated by a factor of 10).

C AM and FM Radio Signals

To enhance your understanding of analog signals, it's worth some effort to understand two types of analog signal that are "close to home," AM and FM radio signals.

AM RADIO SIGNALS

At one time or another, everyone who reads this appendix has listened to an AM (amplitude modulation) radio, the oldest form of transmitting information using radio waves. AM is a way of embedding a low-frequency audio signal in a high-frequency radio signal. The radio signal is transmitted using an antenna, and AM radio receivers are able to extract the audio signal from the transmitted radio signal. The audio signal, in the form of a fluctuating voltage, is then amplified and applied to a loudspeaker. Let's look at an example of those audio and radio frequency (RF) signals.

Figure C-1(a) shows a 440 Hz microphone output audio signal generated by the A key above middle C on a piano keyboard. Figure C-1(b) shows a high-frequency 1.2 MHz (1.2 megahertz) RF sine wave voltage. If we multiply the RF sine wave signal by the low-frequency audio signal, the result is the modulated RF signal shown in Figure C-1(c). Notice that the modulated RF signal's frequency remains at 1.2 Mhz but it's peak-to-valley amplitude fluctuates as time passes. As shown by the dashed-line curve in Figure C-1(d), those amplitude fluctuations are called the *envelope* of the modulated RF signal.

The crucial result here is that the modulated RF signal's envelope is identical to the 440 Hz modulating audio signal in Figure C-1(a). So when the Figure C-1(c) amplitude-modulated RF signal is transmitted as an electromagnetic wave using an antenna, an AM radio receiver can be tuned to 1.2 Mhz and extract the RF envelope audio signal from the modulated radio signal. In the radio receiver, the extracted 440 Hz audio signal is amplified, applied to a loudspeaker, and we then hear the 440 Hz A-key musical note of a piano.

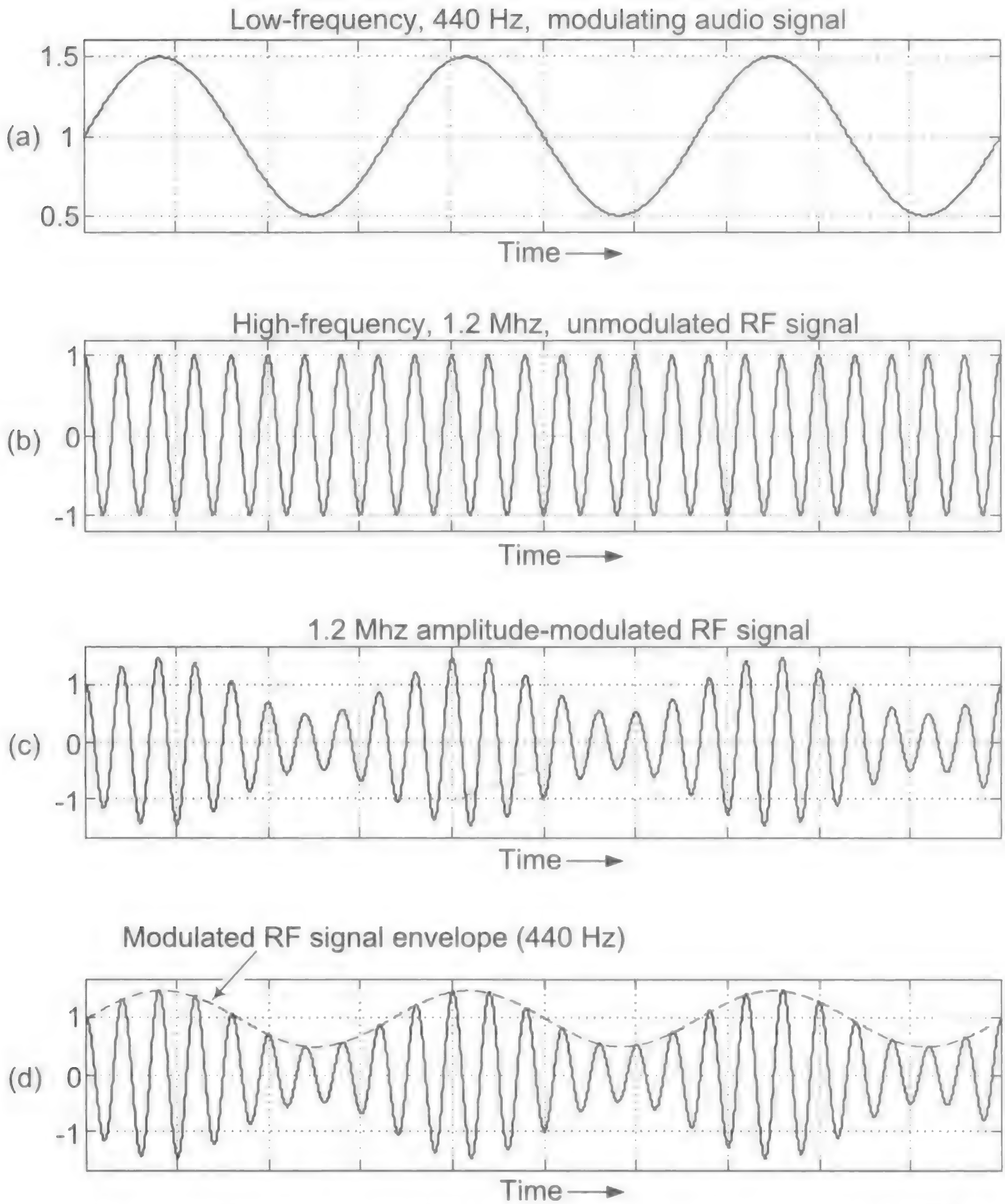


Figure C-1 AM radio signals: (a) 440 Hz audio tone; (b) unmodulated 1.2 Mhz RF (radio frequency) sine wave signal; (c) amplitude-modulated RF signal; (d) RF signal's envelope (dashed curve).

In a similar manner, if the low-frequency modulating audio signal were a voice signal from a microphone, we'd hear human speech from our AM receiver's loudspeaker. As for the spectrum of a broadcast AM signal, that topic is covered in the early part of Chapter 5.

During the second half of the twentieth century, all video (image) signals of commercial analog television were broadcast using AM technology. (Some of you might remember the elaborate television antennas mounted atop every home and apartment.) However, combined with those AM video signals were the television audio signals, which were broadcast using a technique called frequency modulation. That is our next subject.

FM RADIO SIGNALS

To understand the analog signals used in FM (frequency modulation) radio, let's again think about a 440 Hz audio signal from the A key on a piano and a high-frequency 1.2 MHz RF sine wave voltage. Those signals are shown in Figure C-2(a) and in Figure C-2(b).

In FM radio, the frequency of the RF sine wave signal is controlled (modulated) by the 440 Hz audio signal. As shown in Figure C-2(c), the modulated RF signal's frequency is increased when the 440 Hz modulating audio signal has a positive amplitude. And the RF signal's frequency is decreased when the audio signal has a negative amplitude. The result is a high-frequency FM radio signal whose instantaneous frequency depends on the amplitude of the modulating audio signal.

The Figure C-2(c) frequency-modulated RF signal is then transmitted as an electromagnetic wave using an antenna. When an FM radio receiver is tuned to 1.2 Mhz, the receiver generates a signal whose amplitude depends on the instantaneous frequency of the high-frequency FM RF signal. That generated signal is identical to the 440 Hz audio modulating signal. The generated signal is amplified, applied to a loudspeaker, and we hear the 440 Hz A-key musical note of a piano. If the low-frequency modulating audio signal had been a voice signal from a microphone, we'd hear human speech from our FM receiver's loudspeaker.

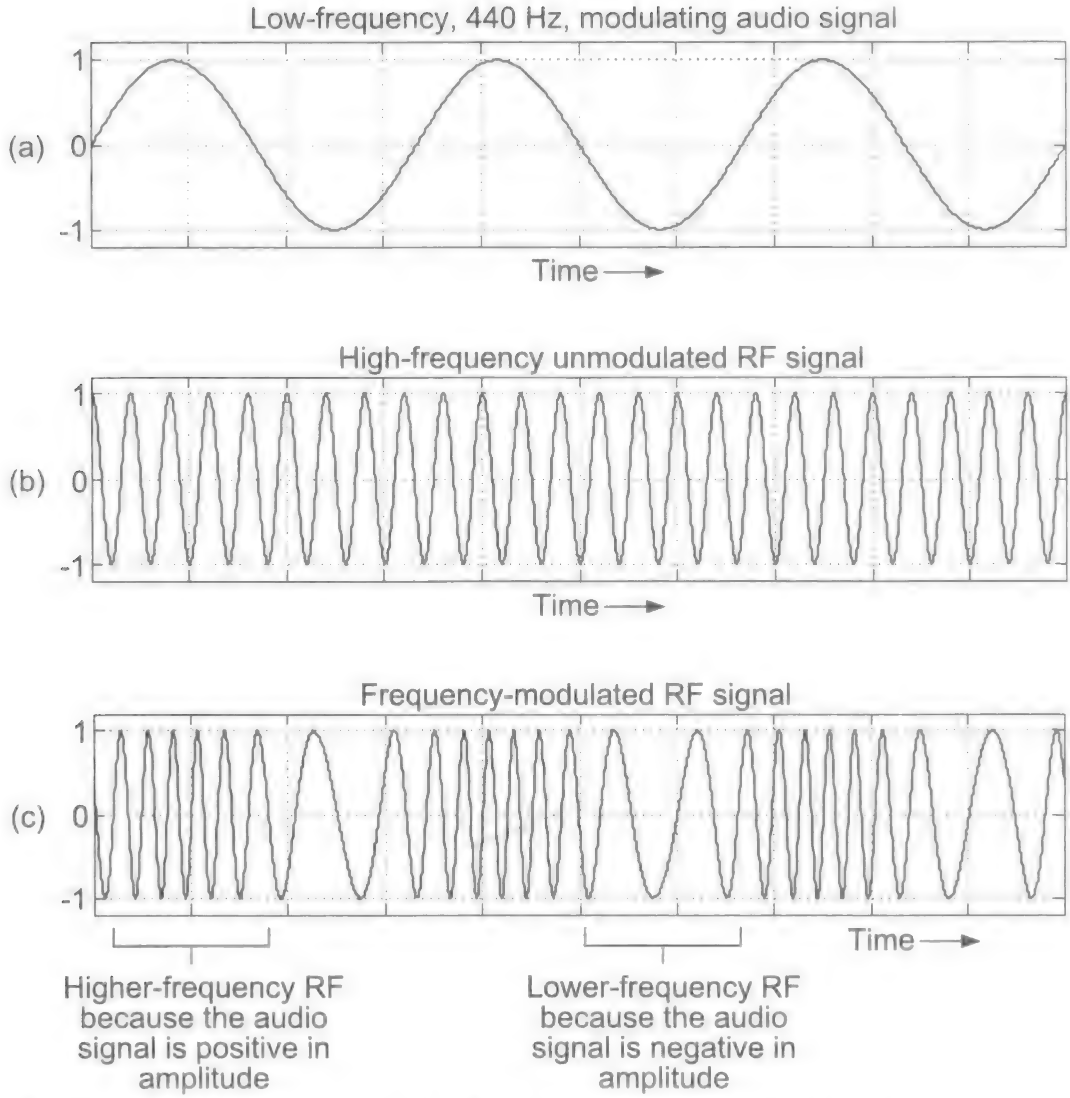


Figure C-2 FM radio signals: (a) 440 Hz audio tone; (b) unmodulated high-frequency RF sine wave signal; (c) frequency-modulated RF signal.

COMPARING AM AND FM RADIO

Table C.1 provides a brief comparison of commercial AM and FM radio as it is used in the United States.

By the Way

In general, history books proclaim Italian experimenter Guglielmo Marconi as the inventor of radio. The fact is, the first person to implement radio transmission and reception was the enigmatic American engineer Nikola Tesla in 1895. In late 1901, with the backing of wealthy benefactors, Marconi became famous for transmitting Morse code signals across the Atlantic Ocean. What wasn't advertised was the fact that Marconi used Tesla-invented oscillator coils for his transatlantic demonstration. Battles over patents raged between Tesla and Marconi for decades. With the drop of the gavel in 1943, a few months after Tesla's death, the U.S. Supreme Court upheld patent number 645,576—Tesla's original radio patent. Sadly, historians have ignored that court decision.

Table C.1 Commercial AM and FM Radio Comparison

Characteristic	AM Radio	FM Radio
When the radio modulation method first became commercially used	1920s	1940s (1960s for stereo)
Hardware complexity of the radio transmitter hardware	Simple	Moderately complex
Hardware complexity of the radio receiver hardware	Simple	Moderately complex
Commercial transmission band	540–1,610 kHz	88–108 MHz
Station-to-station frequency separation	10 kHz	200 kHz
Transmitter power	50–50,000 watts	50–35,000 watts
Transmission audio bandwidth	5 kHz	15 kHz
Stereo (dual channel) capability	No	Yes
Received audio fidelity	Acceptable for voice signals. Poor fidelity for music signals.	Good fidelity for both voice and music signals.
Susceptibility to radio frequency noise	Very susceptible	Very resistant

D Binary Number Formats

In digital signal processing, there are many ways to represent numerical data in computing hardware. These representations, called **binary number formats**, each have their own advantages and shortcomings. The simpler number formats enable us to use uncomplicated hardware designs at the expense of a restricted range of number representation and susceptibility to arithmetic errors. The more elaborate number formats are somewhat difficult to implement in hardware, but they allow us to manipulate very large and very small numbers while providing immunity to many numerical precision problems associated with binary arithmetic. The number format chosen for any given application can mean the difference between processing success and failure—it's where our digital signal processing rubber meets the road.

In this appendix, we'll describe the most common types of binary number formats, and show why and when they're used.

UNSIGNED BINARY NUMBER FORMAT ....

In Chapter 9, we introduced binary numbers and provided the following table to show how sequences of binary digits (0 and 1) could be used to represent decimal numbers.

The number of bits in a binary number is known as its word length—hence 1101 has a word length of four, with the leftmost bit known as the most significant bit (msb), while the rightmost bit is called the least significant bit (lsb).

The binary numbers in Table D.1 are called **unsigned binary numbers** because in their current form, they can only represent positive decimal numbers. For binary numbers to be at all useful in practice, they must be able to represent negative decimal values. There are a number of different ways to represent both positive and negative decimal numbers in binary form and here we present the most popular ways to do so.

Table D.1 First 16 Binary Numbers

Binary Number	Decimal Value
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

SIGN-MAGNITUDE BINARY NUMBER FORMAT........

We can use binary numbers to represent negative decimal numbers by dedicating one of the bits in a binary word to indicate the sign of a number. Let's consider a popular binary number format known as **sign-magnitude**. Here, we assume that a binary word's leftmost bit is a sign bit and the remaining bits represent the magnitude of a number that is always positive. For example, we can say that the 4-bit binary number 0011_2 is equal to $+3_{10}$ and the binary number 1011_2 is equal to -3_{10} as shown in Figure D-1.

Of course, using one of the bits as a sign bit reduces the magnitude of the decimal numbers we can represent. A 4-bit unsigned binary word, for example, can represent 16 different decimal integer values, 0–15_{10}. In the sign-magnitude binary format, a 4-bit word can only represent decimal numbers in the range of -7_{10} to $+7_{10}$.

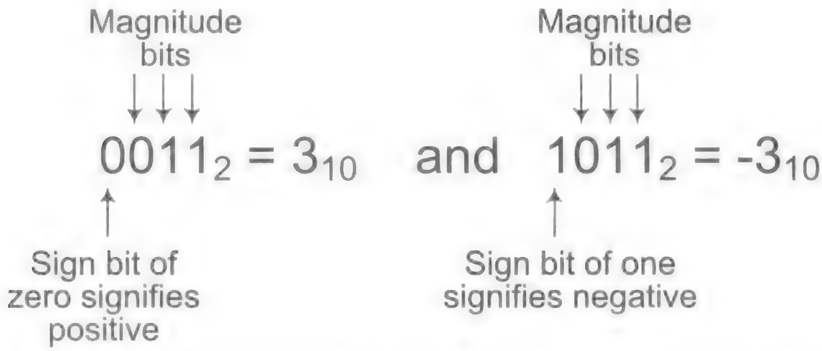


Figure D-1 *Sign-magnitude number format system using the leftmost bit to indicate positive and negative decimal numbers.*

TWO'S COMPLEMENT BINARY NUMBER FORMAT...

Another common binary number scheme is known as the **two's complement** format that also uses the leftmost bit as a sign bit. The two's complement format is the most convenient numbering scheme from a hardware-design standpoint and has been used for decades. It enables computers to perform both addition and subtraction using the same computational hardware. To obtain the negative version of a positive two's complement binary number, we merely complement (change a 1 bit to a 0 bit, and change a 0 bit to a 1 bit) each bit, and add 1 to the complemented word. For example, with 0011_2 representing a decimal 3_{10} in two's complement format, we obtain -3_{10} through the steps shown in Figure D-2.

In the two's complement format, a 4-bit word can represent decimal numbers in the range of -8_{10} to $+7_{10}$. Table D.2 shows 4-bit word examples of sign-magnitude and two's complement binary formats.

$+3_{10}$ in two's complement $\longrightarrow$ 0011

complement of $+3_{10}$ $\longrightarrow$ 1100

add one $\longrightarrow$ +0001

-3_{10} in two's complement $\longrightarrow$ 0011

Figure D-2 Obtaining the negative version of a positive two's complement binary number.

Table D.2 Examples of Binary Number Formats

Sign-Magnitude	Two's Complement	Offset Binary	Decimal Value
0111	0111	1111	7
0110	0110	0110	6
0101	0101	1101	5
0100	0100	1100	4
0011	0011	1011	3
0010	0010	1010	2
0001	0001	1001	1
0000	0000	1000	0
1000	–	–	–0
1001	1111	0111	–1
1010	0110	0110	–2
1011	1101	0101	–3
1100	1100	0100	–4
1101	1011	0011	–5
1110	1010	0010	–6
1111	1001	0001	–7
–	1000	0000	–8

While using two's complement numbers, we have to be careful when adding two numbers that have different word lengths. Consider the case where a 4-bit number is added to an 8-bit number as presented in Figure D-3.

No problem so far. The trouble occurs when our 4-bit number is negative. Instead of adding a $+3_{10}$ to the $+15_{10}$, let's try to add a -3_{10} to the $+15_{10}$ as shown in Figure D-4.

$+15_{10}$ in two's complement ⟶ 00001111

add $+3_{10}$ in two's complement ⟶ +0011

$+18_{10}$ in two's complement ⟶ 00010010

Figure D-3 Adding two positive two's complement numbers that have different word lengths.

$+15_{10}$ in two's complement $\longrightarrow$ 00001111

add -3_{10} in two's complement $\longrightarrow$ +1101

$+28_{10}$ in two's complement $\longrightarrow$ 00011100 $\longleftarrow$ Wrong answer

Figure D-4 Incorrectly adding a positive and a negative two's complement number that have different word lengths.

$+15_{10}$ in two's complement $\longrightarrow$ 00001111

add a sign-extended -3_{10} $\longrightarrow$ +11111101

$+12_{10}$ in two's complement $\longrightarrow$ 100011100 $\longleftarrow$ That's better

Overflow bit is ignored

Figure D-5 Correctly adding a positive and a negative two's complement number that have different word lengths.

The arithmetic error above can be avoided by performing what's called a *sign-extend* operation on the 4-bit number. This process, typically performed automatically in computer hardware, extends the sign bit of the 4-bit negative number to the left making it an 8-bit negative number. If we sign-extend the -3_{10} and then perform the addition, we'll get the correct answer as we see in Figure D-5.

OFFSET BINARY NUMBER FORMAT........................

Another useful binary number scheme is known as the **offset binary** number format. Although this format is not as common as the two's complement binary number format, it still shows up in some hardware. Table D.2 shows offset binary format examples for 4-bit binary words. It may interest the reader that we can convert, back and forth, between the two's complement and offset binary formats merely by complementing (change a 1 to a 0, and change a 0 to a 1) a binary word's most significant bit.

The history, arithmetic, and utility of the many available number formats is a very broad field of study. A thorough and very readable discussion of the subject is given in Donald E. Knuth's *The Art of Computer Programming: Seminumerical Methods*, vol. 2.

ALTERNATE BINARY NUMBER NOTATION..............

As the use of business computers in the 1960s and home computers in the 1970s rapidly expanded, computer programmers grew tired of manipulating long strings of ones and zeros on paper and began to use more convenient ways to represent binary numbers using digits other than a 1 or a 0. The two most popular notational ways to represent binary number are called **octal binary number notation** and **hexadecimal binary number notation**.

Octal Binary Number Notation

The **octal binary number notation** uses a base-8 number system. Converting from binary to octal is as simple as separating the binary number into 3-bit groups starting from the right. For example, the binary number 10101001_2 can be converted to octal format as shown in Figure D-6. In that figure, we use a subscripted 8 to signify an octal number.

The value of using octal notation is that it's easier to write, remember, and verbalize a 3-digit octal number than an 8-digit binary number. The octal number 251_8 is simply easier for programmers to work with than the binary number 10101001_2. Of course, the only valid digits in the octal notation are 0–7. The decimal digits 8 and 9 have no meaning in octal representation.

Hexadecimal Binary Number Notation

Another popular binary format is the **hexadecimal binary number notation** using 16 as its base. Converting from binary to hexadecimal is done by separating the binary number, this time, into 4-bit groups starting from the right. The binary number 10101001_2 is converted to hexadecimal format as shown in Figure D-7.

The peculiar aspect of hexadecimal number notation is that it uses letters to represent decimal values greater than 9. For example, the 4-bit number $1010_2 = 10_{10}$ in Figure D-7 is represented as A16 in hexadecimal notation. Here, we use a subscripted 16 to signify a hexadecimal number.

$$10101001_2 \longrightarrow \underbrace{10}_{2_8}\,|\,\underbrace{101}_{5_8}\,|\,\underbrace{001}_{1_8} = 251_8$$

Figure D-6 Representing an 8-bit binary word with 3 octal digits.

$$10101001_2 \rightarrow \underbrace{1010}_{A_{16}}\,|\,\underbrace{1001}_{9_{16}} = A9_{16}$$

Figure D-7 Representing an 8-bit binary word with 2 hexadecimal digits.

If you haven't seen hexadecimal notation before, don't let the $A9_{16}$ number above confuse you. In this notation, the letters A, B, C, D, E, and F represent the decimal digits 10, 11, 12, 13, 14, and 15, respectively. Table D.3 lists the permissible digit representations in the alternate octal and hexadecimal binary number notations.

Like octal notation, the value of using hexadecimal notation is that it's simply easier to write, remember, and verbalize a 2-digit hexadecimal number than an 8-digit binary number.

Table D.3 The Digits Used in Alternate Binary Number Notations

Binary	Octal	Hexadecimal	Decimal Value
0	0	0	0
1	1	1	1
–	2	2	2
–	3	3	3
–	4	4	4
–	5	5	5
–	6	6	6
–	7	7	7
–	–	8	8
–	–	9	9
–	–	A	10
–	–	B	11
–	–	C	12
–	–	D	13
–	–	E	14
–	–	F	15

Glossary

AC
See alternating current.

ADC
See analog-to-digital converter.

Algorithm
A clearly defined sequence of mathematical steps performed on digital signals and used to perform a given signal processing objective. The processes of digital filtering and the discrete Fourier transform (DFT) are examples of algorithms.

Alias
The frequency of a high-frequency analog sine wave that, when sampled with an analog-to-digital converter, appears as a lower-frequency sine wave in the converted digital signal.

Aliasing
An undesirable effect that can occur when sampling an analog signal. Aliasing occurs when a high-frequency analog signal spectral component falsely appears as a low-frequency spectral component in the sampled digital signal. Aliasing is avoided if the sample rate of the analog-to-digital conversion is greater than twice the highest-frequency spectral component in the input analog signal (Nyquist sampling criterion).

Alternating current (AC)
A term used to describe an analog signal voltage whose amplitude fluctuates over time.

AM
See amplitude modulation.

Amplifier
An electronic circuit, or electronic device, used to increase the amplitude (or power) of an analog signal.

Amplitude
The voltage at any instant in time of a time-domain analog voltage waveform indicating its instantaneous energy. Sometimes engineers use the word *amplitude* to describe the maximum positive value of a time-domain sinusoidal wave.

Amplitude modulation
A radio communication method where the amplitude of a high-frequency sine wave is modulated (controlled) by the amplitude of a low-frequency audio signal. The high-frequency amplitude modulated sine wave can then be transmitted as an electromagnetic wave using an antenna. AM radio receivers are designed to receive and extract the audio signal from the amplitude-modulated radio wave.

Analog filter
A collection of interconnected electronic hardware components that transforms an analog voltage signal (the input) into another voltage signal (the output) having a modified frequency domain spectrum. See lowpass, bandpass, and highpass filters.

Analog signal
In contrast to a digital signal that is a sequence of discrete numbers, an analog signal is an information-carrying continuous signal (typically a voltage) that varies over time and can take on any value between its minimum and maximum values.

Analog-to-digital converter (ADC)
A hardware device that accepts, at its input, an analog voltage signal and produces a periodic stream of numbers. The numbers represent the value of the analog voltage at periodically spaced instants in time. The repetition rate of the instants in time is determined by the frequency (sample rate) of the periodic clock input to the converter.

Anti-aliasing filter
An analog lowpass filter used to limit the bandwidth of an analog signal before that analog signal is applied to an analog-to-digital converter (ADC). Also used to describe an analog lowpass filter used to limit the bandwidth of an analog signal produced by a digital-to-analog converter (DAC).

Attenuation
An amplitude loss, usually measured in dB, incurred by a signal after passing through a digital filter. Filter attenuation is the ratio, at a given frequency, of the signal amplitude at the output of the filter divided by the signal amplitude at the input of the filter.

Audio
Describes signals whose frequency content is in the range of 50 Hz to 15 kHz. An audio voltage signal, applied to the terminals of a loudspeaker, will generate air pressure waves that can be heard as sound by human beings.

Averager
See moving averager.

Bandpass filter
A filter, as shown in Figure G-1, that passes one frequency band, from frequency f_1 to frequency f_2, and attenuates frequencies above and below that band.

Bandstop filter
A filter that rejects (attenuates) one frequency band, from frequency f_1 to frequency f_2, and passes both a lower- and a higher-frequency band. Figure G-2 depicts the frequency response of an ideal band reject digital filter.

Bandwidth
The frequency range over which a signal contains significant spectral energy or the frequency width of the passband of a filter. See passband.

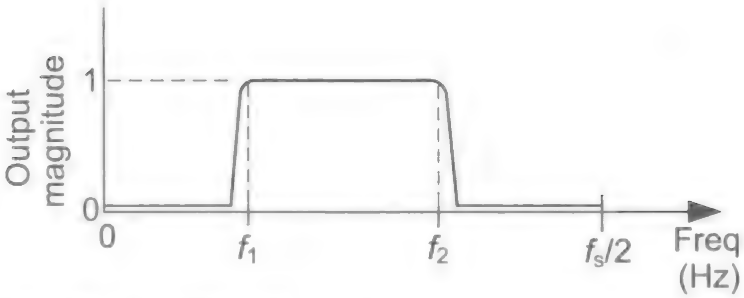


Figure G-1 Bandpass filter.

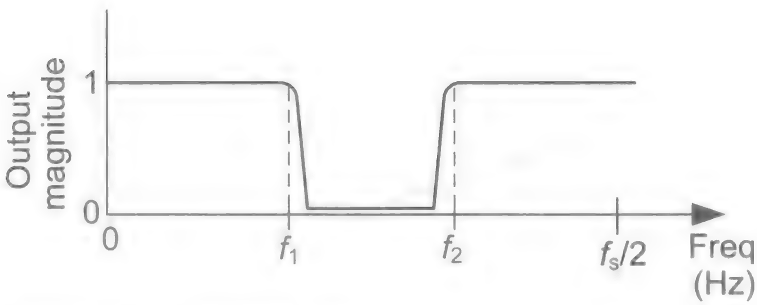


Figure G-2 Bandstop filter.

Base-2 number system
See binary number system.

Base-10 number system
Our familiar system of counting that has ten different digits, 0 through 9. Chapter 9 discusses the base-10 number system and other numbers systems used in digital signal processing.

Binary number system
A system of counting that has only two digits, 0 and 1. All of the arithmetic operations that we perform in our traditional decimal number system, such as addition, subtraction, multiplication, and division, can be performed in the binary number system. The binary number system is used because it is the most inexpensive and reliable way to perform arithmetic operations using electronic components.

Bit
A bit (short for *Binary digIT*) is the smallest unit of information in a binary number. A bit is a single binary digit of either 0 or 1.

Broadcast
The process of radiating an information-carrying analog electromagnetic signal using an antenna.

Byte
A sequence of 8 binary bits.

Carrier frequency
The frequency of the center of the bandwidth of a broadcasted radio signal.

Cascaded filters
The implementation of a filtering *system* where multiple individual filters are connected in a series. That is, the output of one filter drives the input of the following filter as illustrated in Figure G-3.

CD
See compact disc.

Cell phone
Short for cellular phone: a mobile telephone.

Center frequency
The frequency located at the midpoint of a bandpass filter. Figure G-4 shows the f_o center frequency of a bandpass filter. Center frequency is also used to specify the frequency located at the midpoint of a band of signal spectral energy.

Chip
Slang for a single integrated circuit.

Circuit
An organized interconnection of electronic hardware devices that accomplishes an electrical objective.

Clock signal
A square wave voltage inside a hardware device containing digital signals, such as a computer or a cell phone, used to synchronize the operation of various electronic circuits.

Compact disc (CD)
A plastic disc with a thin metal surface on one side, 4.7 inches in diameter, used to store both binary digital data files as well as binary digital music signals. CDs are able to store 0.7 gigabits of binary data.

Continuous signals
See analog signal.

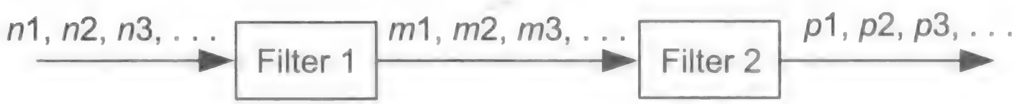


Figure G-3 Cascaded filters.

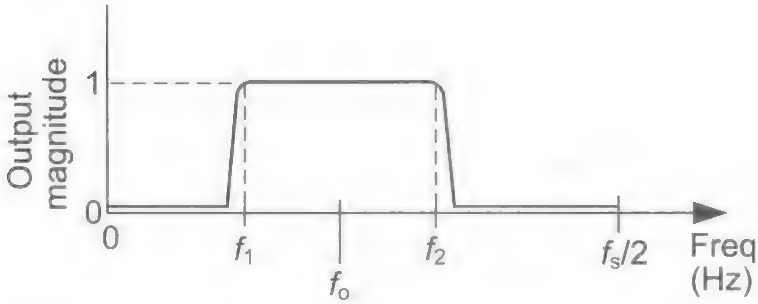


Figure G-4 Center frequency of a bandpass filter.

Cosine wave
A sinusoidal wave form whose initial value (at zero time) is shown by the solid-line curve in Figure G-5.

Cutoff frequency
The upper passband frequency for lowpass filters, and the lower passband frequency for highpass filters. Figure G-6 illustrates the f_{co} cutoff frequency of a lowpass filter.

Cycles per second
The unit of measure for frequency.

DAC
See digital-to-analog converter.

dB
See decibel.

DC
Acronym for direct current. A long-standing technical term used to describe a signal (either an analog voltage or a discrete sequence of numbers) whose amplitude remains constant as time passes.

DC voltage
An analog voltage whose amplitude remains constant as time passes.

Decibel
A unit of attenuation, or gain, used to express the relative voltage or power difference between two signals. Appendix B discusses decibels in detail.

Decimal numbers
The number system we use every day that includes 10 digits, 0, 1, 2, 3, … 9.

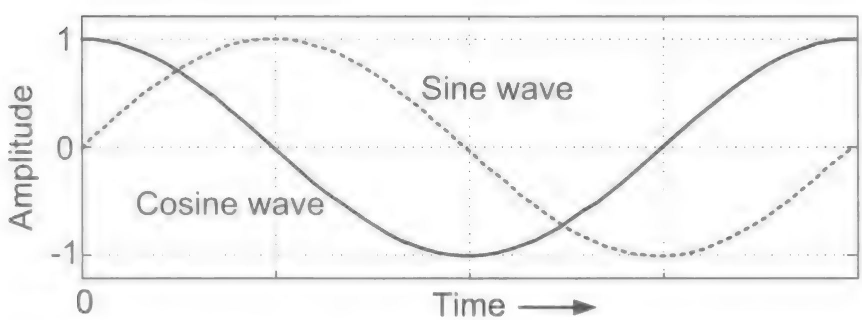


Figure G-5 Cosine wave.

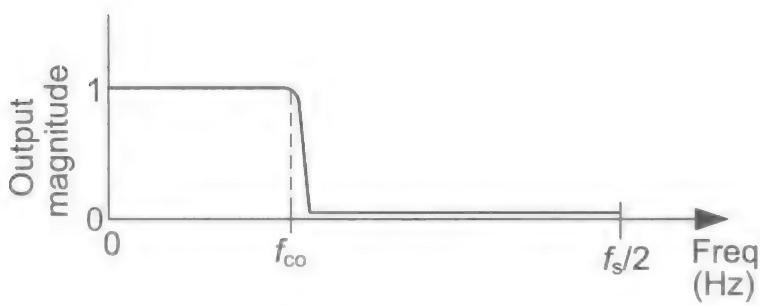


Figure G-6 Cutoff frequency of a lowpass filter.

Decimation
Decreasing the sample rate of a digital signal.

DFT
See discrete Fourier transform.

Digital filter
A computational process, or algorithm, that transforms a discrete time-domain sequence of numbers (the input) into another discrete sequence of numbers (the output) having a modified frequency domain spectrum. See lowpass, bandpass, and highpass filters.

Digital number
A number used, or displayed, by a piece of electronic hardware. The time shown on the face of a digital clock is a digital number, as are the numbers on a computer keyboard.

Digital signal
An information-carrying discrete sequence of numbers where each numerical value can only be one of a limited set of possible values. This is the predominate definition of *digital signal* used in this book. Also refers to an analog voltage inside electronic equipment that fluctuates between one of two voltage values.

Digital signal processing
The numerical processing of signal data samples.

Digital signal processor
An integrated circuit specifically designed to efficiently perform arithmetic operations on sequences of numerical signal data samples.

Digital-to-analog converter
A hardware device that accepts, at its input, digital sample values and produces an output analog voltage signal.

Digital video disc (DVD)
A metallic disc, 4.7 inches in diameter, used to store both digital data files as well as digital music signals. DVDs are able to store 4.7 gigabits of binary data.

Direct current
A long-standing technical term used to describe a signal (either an analog voltage or a discrete sequence of numbers) whose amplitude remains constant over time.

Discrete Fourier transform
The mathematical process performed on digital signal sequences (the input) to produce a sequence of numbers representing the spectral (harmonic) content of those input time sequences.

Downsample
See decimation.

DSP
See digital signal processing.

DSP
See digital signal processor.

Fast Fourier transform (FFT)
A specialized algorithm (a sequence of mathematical steps) to *very* efficiently perform discrete Fourier transforms (DFTs). By "efficiently" we mean with a *drastically* reduced number of mathematical operations, which is a very good thing indeed.

Filter
A hardware device (for analog signals) or arithmetic process (for digital signals), used to modify the spectral content of a time-domain signal.

Finite impulse response filter (FIR)
Defines a class of digital filters that have exactly linear-phase behavior.

FIR
See finite impulse response filters.

FM
See frequency modulation.

Frequency
A measure of how many complete cycles of a periodic wave repeat in the time period of one second. The typical units of measure of frequency are hertz (Hz). For convenience in mathematical analysis (algebra), frequency is sometimes measured in radians per second. See radians per second.

Frequency domain
An adjective meant to specify the spectral content of a signal.

Frequency modulation
A radio communication method where the frequency of a high-frequency sine wave is modulated (controlled) by the amplitude of a low-frequency audio signal. The high-frequency FM sine wave can then be transmitted as an electromagnetic wave using an antenna. FM radio receivers are designed to receive and extract the audio signal from the frequency modulated radio wave.

Frequency response
A frequency domain description of how a filter interacts with input signals. The bold solid curve in Figure G-7(a) is the frequency response of a digital lowpass filter having a cutoff frequency of f_{co} Hz. Figure G-7(b) depicts the frequency response of a digital highpass filter.

Gain
The amount of amplification accomplished by an amplifier circuit. For example, a gain of 4 means the amplifier output signal's amplitude is four times as great as the input signal's amplitude.

Gb
An acronym for gigabits, 1,073,741,824 bits of binary data.

GB
An acronym for gigabytes, 1,073,741,824 bytes = 8 × 1,073,741,824 = 8,589,934,592 bits of binary data.

Ghz
An acronym for the frequency of 1 gigahertz (1,000,000,000 cycles per second).

Ground
A point, or node, in an electrical circuit to which all voltages in the circuit are referred.

Half-band filter
A type of FIR digital filter where the transition region is centered at one-quarter of the sampling rate, or $f_s/4$. Due to their frequency domain symmetry, half-band filters are often used in decimation schemes because half their time domain coefficients are zero. This reduces the number of necessary multiplication operations needed to produce each filter output sample.

Harmonic distortion
An inadvertent and undesirable modification of the shape of a time signal that produces unwanted spectral components in that time signal.

Harmonics
The undesirable spectral components in a time signal that cause an inadvertent distortion of

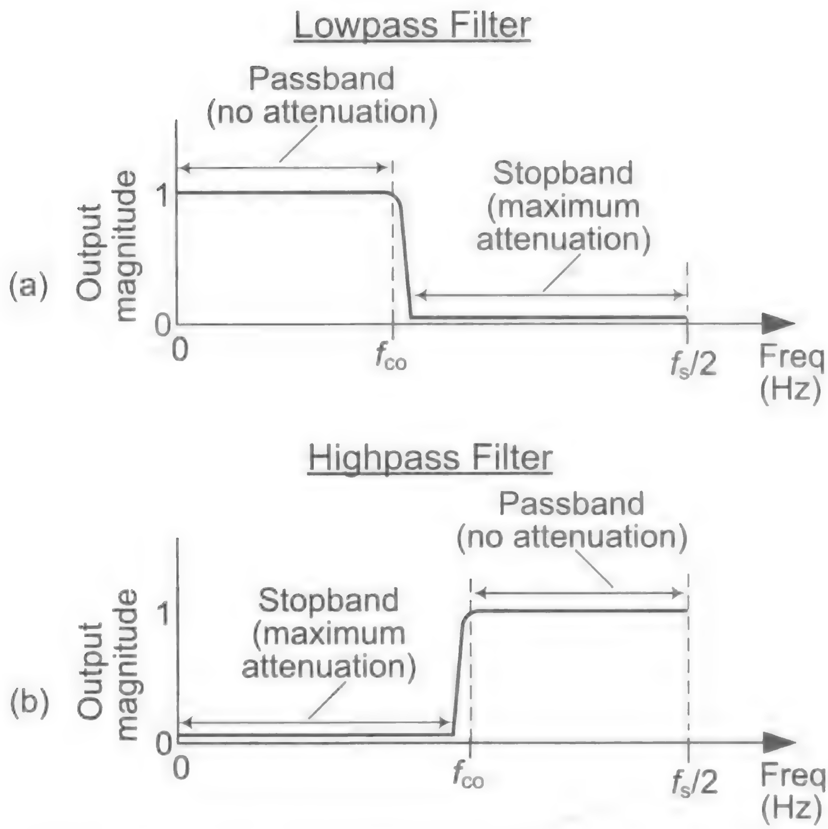


Figure G-7 Filter frequency responses: (a) lowpass filter; (b) highpass filter.

the shape of the time signal. Also used to describe the high-frequency spectral components contained in periodic time signals (such as square or triangular waves).

HDTV

See high-definition television.

Hertz (Hz)

A measure of frequency equivalent to cycles per second (oscillations per second).

Hexadecimal binary numbers

A convenient way for computer programmers to represent binary numbers using digits other than a 1 or 0. See Appendix D.

High-definition television

An all-digital system for transmitting a TV signal with far greater resolution than the older analog TV signals. A high-definition television (HDTV) can display several resolutions (up to two million pixels versus an older television's 360,000). HDTV uses 24 binary bits to define a pixel's color, which provides improved color rendition.

High fidelity

A term used by audio enthusiasts that refers to high-quality audio sound. By high-quality we mean, for example, how the audio from an expensive FM stereo system sounds compares to the audio we hear over a cell phone.

Highpass filter
A filter that passes high frequencies and attenuates low frequencies as shown in Figure G-7(b). We've all experienced a kind of highpass filtering in our living rooms. Notice what happens when we turn up the treble control (or turn down the bass control) on our home stereo systems. The audio amplifier's normally flat frequency response changes to a kind of analog highpass filter giving us that sharp and *tinny* sound as the high-frequency components of the music are accentuated.

IIR
See infinite impulse response filter.

Impulse response
A digital filter's time domain output sequence when the input is a single unity-valued sample (impulse) preceded and followed by zero-valued samples. A digital filter's frequency domain response can be calculated by performing the discrete Fourier transform (DFT) of the filter's time domain impulse response.

Infinite impulse response (IIR) filter
Defines a class of digital filters that are not guaranteed to be stable and always have nonlinear phase responses. Infinite impulse response (IIR) filters have a much steeper transition region roll-off (superior performance) than digital finite impulse response (FIR) filters.

Integers
Positive or negative whole numbers such as 23, –57, or 99.

Integrated circuit
A miniaturized collection of interconnected (integrated) transistors encapsulated in a small plastic or ceramic block.

Interpolation
Increasing the sample rate of a digital signal.

JPEG
Industry-standard electronic image file compression method. JPEG stands for Joint Photographic Experts Group.

Kb
An acronym for kilobits, 1,024 bits of binary data.

KB
An acronym for kilobytes, 1,024 bytes = $8 \times 1{,}024 = 8{,}192$ bits of binary data.

kHz
An acronym for the frequency of 1 kilohertz (1,000 cycles per second).

Ksps
An acronym for kilosamples per second, the rate at which an analog signal can be sampled, measured in thousands of samples per second.

Least significant bit
The rightmost bit in a binary word (sequence of binary bits).

Linear phase filter
A filter that exhibits a constant change in phase angle (degrees) over frequency. The resultant filter phase plot versus frequency is a straight line. As such, a linear phase filter's group delay is a constant. In order to preserve the integrity of their information-carrying signals, linear phase is an important criteria for filters used in cell phones and other wireless systems.

Loudspeaker
A hardware device that converts an electrical voltage signal to air pressure waves (sound).

Loudspeakers operate over the frequency range of 50 Hz to 15 kHz.

Lowpass filter
A filter that passes low frequencies and attenuates high frequencies as shown in Figure G-7(a). By way of an example, we experience lowpass filtering when we turn up the bass control (or turn down the treble control) on our home stereo systems, giving us that dull muffled sound as the low frequency components of the music are intensified.

Mb
An acronym for megabits, 1,048,576 bits of binary data.

MB
An acronym for megabytes, 1,048,576 bytes = 8 × 1,048,576 = 8,388,608 bits of binary data.

Mhz
An acronym for the frequency of 1 megahertz (1,000,000 cycles per second).

Microchip
Slang for a single integrated circuit.

Microphone
A hardware device that converts air pressure waves (sound) to an electrical voltage signal. Microphones operate over the frequency range of 50 Hz to 15 kHz.

Mixing
For audio signals, mixing is the process of adding two or more signals to create a composite audio signal. For signals in the radio frequency range, mixing is the process of multiplying one signal by a second signal. This radio frequency mixing is used to produce AM broadcast radio signals.

Most significant bit
The leftmost bit in a binary word (sequence of binary bits).

Moving averager
An arithmetically simple digital lowpass filtering process where a fixed number of successive input signal sample values are averaged to produce a sequence of filter output samples. See Chapter 8 for an example of a moving averager filtering process.

MPEG
Industry-standard electronic video file compression method. MPEG stands for Motion Picture Experts Group.

Msps
An acronym for megasamples per second. The sample rate, with which an analog signal can be sampled, measured in millions of samples per second.

Nibble
A sequence of 4 binary bits. (Half a *byte*.)

Noise
The uncontrollable amplitude fluctuations of a time-domain signal as shown in Figure G-8(b). Noise is random and carries no useful information. The presence of noise makes it difficult to measure important signal parameters, and it is typically considered to be undesirable.

Nonrecursive filter
A digital filter implementation where no filter output sample is retained for later use in computing a future filter output sample.

Nyquist sampling criterion
A digital signal processing rule that states: To avoid undesirable frequency aliasing (digital

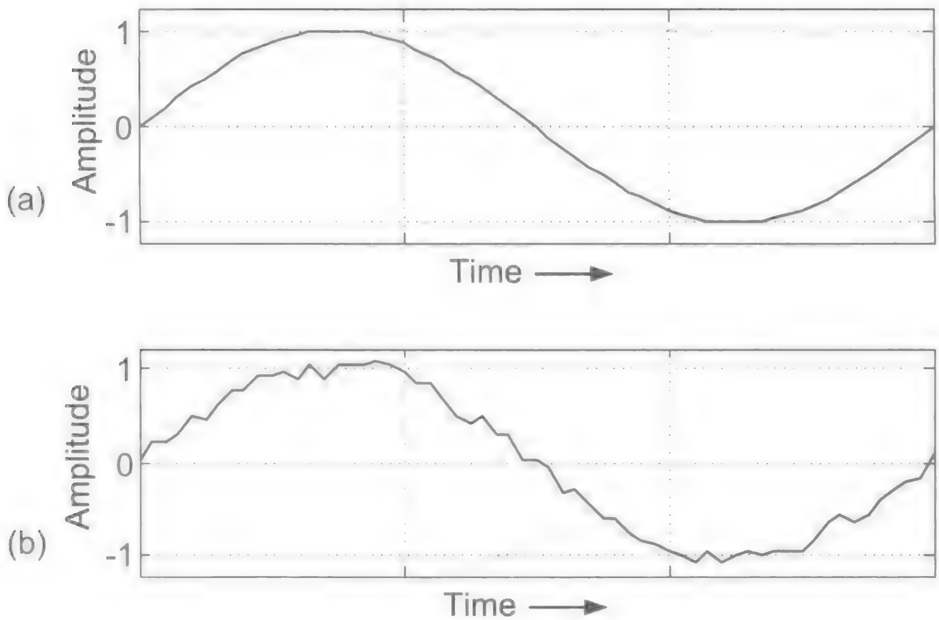


Figure G-8 *Noise: (a) a noise-free analog sine wave; (b) a noise-contaminated analog sine wave.*

signal distortion), the sample rate of analog-to-digital conversion must be greater than twice the highest frequency spectral content in the input analog signal. Named in honor of American engineer Harry Nyquist. Also called the Shannon-Nyquist sampling theorem.

Octal binary numbers
A convenient way for computer programmers to represent binary numbers using decimal digits other than a 1 or a 0. See Appendix D.

Offset binary numbers
A method of using binary numbers to represent both positive and negative decimal numerical values.

Oscilloscope
A piece of electronic equipment that accepts analog voltages as inputs and displays those voltage waveforms as two-dimensional plots on a display screen. The vertical axis of the plot is voltage level and the horizontal axis of the display is time (seconds).

Passband
The frequency range over which a filter passes input signal energy as depicted in Figure G-7.

Passband ripple
Fluctuations in the passband amplitude response of a filter, as shown in Figure G-9.

PC board
See printed circuit board.

Period
The period of time, measured in seconds, that it takes a periodic signal to complete one oscillation.

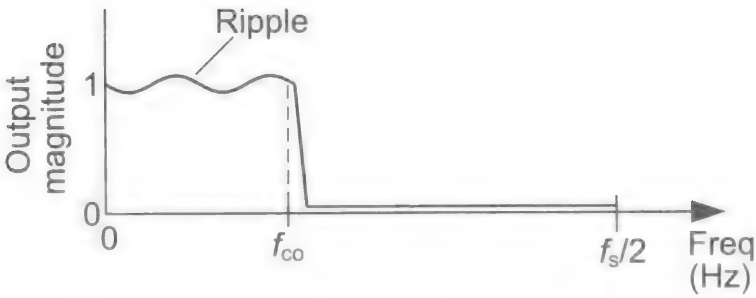


Figure G-9 Filter passband ripple.

Periodic wave

An analog, or digital, signal whose time-domain waveform repeats as time passes.

Phase response

The difference in phase, at a particular frequency, between an input sine wave and the output sine wave at that frequency. The phase response, sometimes called *phase delay*, is usually depicted using a curve showing the filter's phase shift versus frequency.

Pixel

Short for *picture element*. A single point (or dot) of an image such as the image on a computer screen, digital camera photo, or television screen. A 5 megapixel digital camera takes pictures comprising 5,000,000 individual pixels (colored dots). Those pixels come together to form an image. See high-definition television.

Printed circuit board

A rectangular plastic or fiberglass board with conductive copper lines printed or etched on the board. Electronic components are mounted on the board and the conductive lines (traces) connect the components to form a working circuit. The motherboard in your home computer is a printed circuit board. Printed circuit boards avoid the need to hand-solder electric wires between electronic components as was done decades ago.

Quantization

To limit the possible values of a number to a discrete set of values of a specified precision. For example, quantizing the numbers 17.3, –26.88, and 52.13 to integers means to convert those numbers to 17, –26, and 52. As related to analog-to-digital converters, quantization is the process whereby the continuous range of an analog input signal's values is divided into nonoverlapping voltage subranges. When an input analog signal value resides within a given voltage subrange, the converter output provides the corresponding unique, discrete binary numerical value.

Radians per second

An angular measure of how often a periodic wave repeats in the time period of one second. Primarily used in mathematical analysis (algebra). One 360-degree cycle is equal to 2π radians. One radians is roughly equal to 57.2 degrees.

Radio frequency
Ranges of frequency bands of radiated sinusoidal electromagnetic waves:

- ultra-low frequency (ULF): DC–30 Hz
- extremely low frequency (ELF): 30–300 Hz
- voice-grade channel (VGC): 300–3400 Hz
- very low frequency (VLF): 3–30 kHz
- very high frequency (VHF): 30–300 MHz
- ultra-high frequency (UHF): 300–3000 MHz

Recursive filter
A digital filter implementation where current filter output samples are retained for later use in computing future filter output samples.

RF
See radio frequency.

Sample
A single number, a single element, of a digital signal's sequence of numbers. The word *sampling* refers to the process of converting an analog signal to a digital signal (a discrete sequence of numbers) using an analog-to-digital converter.

Sample frequency
See sample rate.

Sample rate
The repetition rate, measured in Hz, of the clock used to initiate the conversion of an analog signal to a digital sequence of numbers by an analog-to-digital converter (ADC). The sample rate is the reciprocal of the time period between samples of a digital signal (a discrete sequence).

Sample rate conversion
The process of either decreasing (by way of decimation) or increasing (by way of interpolation) the sample rate of a digital signal.

Scientific notation
A convenient and precise way for engineers and scientists to write very large and very small numbers. Appendix A discusses scientific notation in detail.

Signal-to-noise ratio
A number equal to the ratio of the power of the desired signal divided by the power of an undesired random noise signal that may be contaminating the desired signal. The larger the signal-to-noise ratio value, the better. This number is often expressed in units of dB.

Sign-magnitude binary numbers
A method of using binary numbers to represent both positive and negative decimal numerical values.

Sine wave
A sinusoidal waveform whose initial value (at zero time) is zero as shown by the dashed-line curve in Figure G-5.

Sinusoidal waves
A generic term meaning either a sine wave or a cosine wave.

SNR
See signal-to-noise ratio.

Spectrum
The frequency content of an analog or digital signal. The combination of sinusoidal waves of different frequencies that make up a signal.

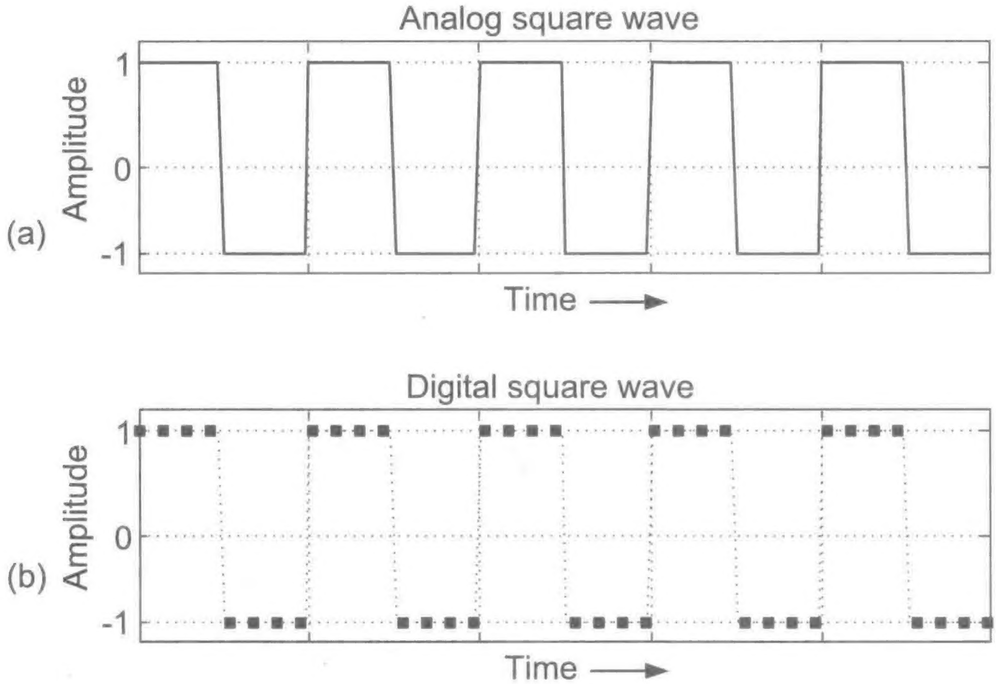


Figure G-10 *Square waves: (a) analog square wave; (b) digital square wave sequence.*

Spectrum analysis

The process of measuring the frequency content of an analog or digital signal.

Spectrum analyzer

Electronic equipment that accepts analog voltages as inputs and displays signal spectra as two-dimensional plots on a display screen. The vertical axis of the plot is signal power level and the horizontal axis of the display is frequency (Hz).

Square wave

A bi-level time-domain waveform as shown in Figure G-10.

Stopband

The band of frequencies attenuated by a digital filter. Figure G-7 shows the stopbands of a lowpass and a highpass filter.

Stopband attenuation

The reduction in amplitude, at the output of a filter, of spectral components residing in the stopband of that filter. Stopband attenuation is typically measured in decibels as shown for a lowpass filter in Figure G-11. In that figure, the stopband attenuation is 40 decibels (dB), which means that the amplitudes of filter input spectral components in the stopband frequency range have been reduced by a factor of 100 (see Appendix B).

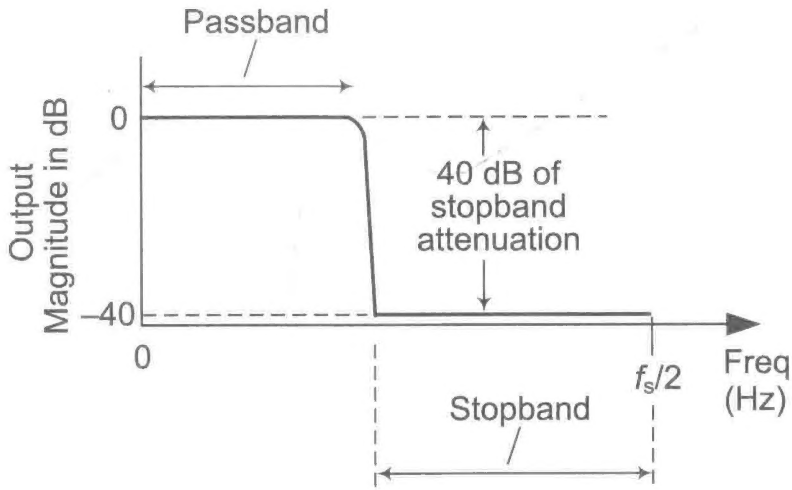


Figure G-11 Lowpass filter stopband attenuation.

Stopband suppression
See stopband attenuation.

Tachometer
A transducer used to measure the rate of revolution, measured in revolutions per second (RPM), of a mechanical shaft.

Time domain
An adjective meant to specify how a signal's amplitude varies as time passes.

Trace
A thin strip of copper conductor deposited on a printed circuit board.

Transceiver
A communication device containing both a receiver and a transmitter. A cell phone is a transceiver.

Transistor
A basic solid-state (silicon) control device that permits or prevents current flow between two electric terminals, based on the voltage or current delivered to a third terminal. Transistors can be used as high-speed switches, either turned on or turned off, or as amplifiers, as in guitar amplifiers.

Transition region
The frequency range over which a filter's frequency response transitions from the passband to the stopband. Figure G-12 illustrates the transition region of a lowpass filter. Transition region is sometimes called the *transition band.*

Transversal filter
In the field of digital filtering, transversal filter is another name for an FIR filter. See finite impulse response filter.

Triangular wave
A time-domain analog waveform as shown in Figure 2-11.

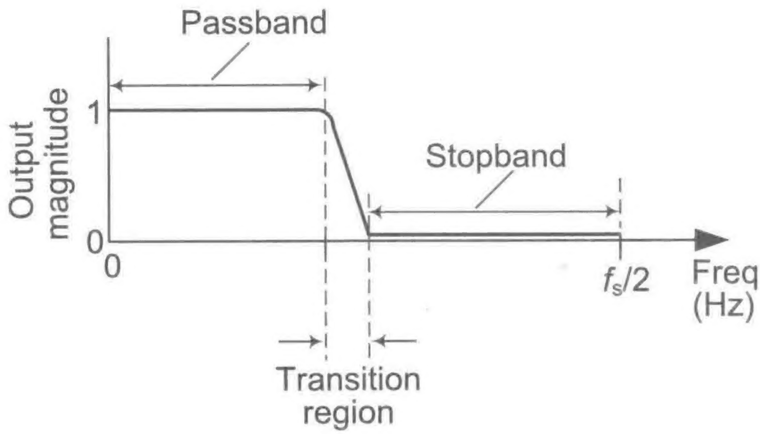


Figure G-12 Lowpass filter transition region.

Two's complement binary numbers

A method of using binary numbers to represent both positive and negative decimal numerical values.

Unsigned binary numbers

A method of using binary numbers to represent positive decimal numerical values.

Upsample

See interpolation.

Voltage

The electric pressure, or potential difference, that causes current (electrons) to flow within an electric circuit. Current flow can do work for us, such as turn an electric motor or generate visible light using a lightbulb. There are two types of voltages: AC (alternating current) voltage, whose voltage amplitude varies as time passes, such as a sine or cosine wave voltage; and DC (direct current) voltage whose voltage amplitude remains constant, such as the voltage of a car battery.

Waveform

Refers to any fluctuating curve shown in a two-dimensional plot having time displayed on the horizontal axis. For example, Figure 2-3(b) is a voltage waveform.

Word length

The number of bits in a binary word. For example, the binary data word 101101 has a word length of 6.